(1)

AD-A172 420

STUDY OF THE EFFECTS OF VIBRATION ON INERTIAL NAVIGATION SYSTEM ACCURACY

THESIS

Donald J. Kocian
Captain, USAF

AFIT/GE/ENG/86J-2

DTIC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DTIC
S ELECTE D
OCT 1 1986
B

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

86 10 01 204

AFIT/GE/ENG/86J-2

STUDY OF THE EFFECTS OF VIBRATION ON INERTIAL NAVIGATION SYSTEM ACCURACY

THESIS

Donald J. Kocian
Captain, USAF

AFIT/GE/ENG/86J-2

DTIC
ELECTE
OCT 1 1986
S D
B

Approved for public release; distribution unlimited

AFIT/GE/ENG/86J-2

STUDY OF THE EFFECTS OF VIBRATION ON INERTIAL NAVIGATION SYSTEM ACCURACY

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Donald J. Kocian, B.S.E.E.
Capt USAF

June 1986

Approved for public release; distribution unlimited

Preface

The purpose of this study was to examine the effects of airframe vibration on the accuracy of an inertial navigation system. The was done by developing models for the vibration and using an existing system model in a Monte Carlo simulation of the system error equations.

I would like to thank Dr Robert Edwards, Dr Peter Maybeck, Dr Robert Fontana, and Mr Stanton Musick for acting as my thesis advisors and for their generous assistance without which this study could not have been accomplished. I would also like to thank my wife, Kathy, and daughter, Kimberly, whose infinite patience was invaluable.

✓

QUALITY INSPECTED
4

A-1

Table of Contents

Page

Preface ii

List of Figures v

List of Tables vii

Abstract viii

I. Introduction 1

- Background 1
- Problem 1
- Scope 2
- Development 3

II. Error Model Development 6

- Basic Error Differential Equations 6
- Coordinate Systems 7
- Altitude Channel Mechanization 13
- Gyro Error Model 15
- Accelerometer Error Model 17
- Gravity Error Model 17
- Barometric Altimeter Error Model 20
- Truth Model for Zero Vibration 20
- Vibration Environment 33
- Experimental Integrated Effect Vibration Model 42
- Initial Conditions 46
- Flight Profile 50
- Mission Profile 50
- Summary 51

III. Software 54

- SOFE 54
- SOFEPL 54
- PROFGEN 55
- Summary 57

IV. Simulation Results 58

- Program Validation 58
- Vibration Induced Errors 62

Comparison of Errors Due to Vibration and Total System Errors 73

V. Conclusions and Recommendations. 82

Conclusions 82
Recommendations 83

Bibliography 86

Appendix A: SOFE. 88

Appendix B: PROFGEN 90

Appendix C: SOFE Subroutines for 50-State Model Without Vibration 92

Appendix D: SOFE Subroutines for Sinusoidal Series Vibration Model 99

Appendix E: SOFE Subroutines for 50-state Model With Integrated Effect Vibration Model . 103

Vita 112

List of Figures

Figure Page

1. F Matrix of the INS General Differential Equations. 10
2. Body and Navigation Coordinate Frames. 12
3. F Matrix for the 50 State SIGN-III Error Model . . 21
4. X-Axis Measured Vibration. 34
5. X-Axis Model Vibration 34
6. Y-Axis Measured Vibration. 35
7. Y-Axis Model Vibration 35
8. Z-Axis Measured Vibration. 36
9. Z-Axis Model Vibration 36
10. N. Attitude Error for I.C. of x(2)=1 arcmin. . . . 59
11. Latitude Error for I.C. of x(2)=1 arcmin 59
12. Longitude Error for I.C. of x(8)=1 arcmin. 60
13. East Attitude Error for I.C. of x(8)=1 arcmin. . . 60
14. Latitude Error for I.C. of x(4)=1 ft/sec 61
15. East Velocity Error for I.C. of x(8)=1 ft/sec. . . 61
16. Latitude Error for Sinusoidal Method 63
17. Latitude Error for Integrated Effect Method. . . . 63
18. Longitude Error for Sinusoidal Method. 64
19. Longitude Error for Integrated Effect Method . . . 64

20. Altitude Error for Sinusoidal Method 65
21. Altitude Error for Integrated Effect Method. . . . 65
22. Up Tilt Error for Sinusoidal Method. 66
23. Up Tilt Error for Integrated Effect Method 66
24. East Tilt Error for Sinusoidal Method. 67
25. East Tilt Error for Integrated Effect Method . . . 67
26. Latitude Error Due to Vibration. 74
27. Latitude Error From Non-Vibration Sources. 74
28. Longitude Error Due to Vibration 75
29. Longitude Error From Non-Vibration Sources 75
30. Vert. Vel. Error Due to Vibration. 76
31. Vert. Vel. Error From Non-Vibraton Sources 76
32. North Tilt Error Due to Vibration. 77
33. North Tilt Error From Non-Vibraton Sources 77
34. East Tilt Error Due to Vibration 78
35. East Tilt Error From Non-Vibration Sources 78
36. Up Tilt Error Due to Vibration 79
37. Up Tilt Error From Non-Vibration Sources 79

List of Tables

Table Page

1. State Definitions for the Fundamental Error Matrix 9
2. Notation Used in Figures 1 and 3 11
3. Model for Gravity Variations 20
4. Notation Used in Figure 3 27
5. Error State Definitions 28
6. SIGN-III Noise Density Matrix, Non-zero Elements . 31
7. SIGN-III Error Source Statistics 32
8. Flight Profile 53
9. PROFGEN Output 56
10. RSS Errors Due to Sinusoidal Vibration 70
11. Comparison of Study Results 70

AFIT/GE/ENG/86J-2

Abstract

This study examines the effects of airframe vibration on the accuracy of a strapdown inertial navigation system. A stochastic model of the system error equations is included, as are two models of airframe vibration. Software subroutines for model implementation in SOFE are included.

A representative C-130A flight profile was developed using a flight trajectory generator, PROFGEN. The system errors induced in the inertial navigation by simulating this mission are included as are those caused by vibration. Vibration induced errors were found to be very small and orders of magnitudes smaller than those caused by other error sources.

STUDY OF THE EFFECTS OF VIBRATION ON THE ACCURACY OF AN INERTIAL NAVIGATION SYSTEM

I. Introduction

Background

Inertial navigation systems have been in use for many years. As the technology of these systems has changed and matured, many problems have been solved and the capabilities of inertial navigation systems have increased greatly. However, as the capabilities of inertial navigation systems improve, new missions emerge which put even greater demands on the system. Error sources which previously were small enough to be ignored, suddenly become significant.

Problem

The Military Airlift Command (MAC) is tasked with providing airlift to all services of the United States armed forces. In order to meet the needs of the services, MAC requires the ability to fly long missions, including low-level flight, without external aiding of their inertial navigation systems. This requires the inertial navigation system to maintain low position and velocity errors which they are able to do in the relatively benign environment of high altitude cruise. However, there are indications that the more severe vibrational environment at low altitude, caused by greater atmospheric turbulence and increased air density, may cause intolerable degradation of inertial navigation system performance.

The problem undertaken in this study will be to analyze the effects of vibration upon the inertial navigation system accuracy. A representative transport mission profile, to include both high and low altitude flight, will be used to excite the low frequency system errors.

The probabilistic approach undertaken here utilizes stochastic models to account for uncertainities in the inertial navigation system. Stochastic process and modern estimation theories will be used to characterize the initial conditions, forcing functions, and the resulting system outputs.

Scope

The focus of this study will be to identify the relative effect of vibration on overall system performance. This study will use an error model of the Honeywell SIGN-III strapdown inertial navigation system. The model used is one which was developed by Widnall and Grundy (Ref 3). A flight profile will be generated based on the performance characteristics of the Lockheed C-130A aircraft. This aircraft was chosen because it is the most common aircraft in the MAC fleet. It provides a more severe vibrational environment than either the C-141 or the C-5A, and some flight test data is available on C-130 vibration levels.

Development

The initial portion of this study entails the implementation of the SIGN-III error model to perform the error analysis. The model equations will be implemented in a digital simulation program called SOFE which was developed by Stanton H. Musick (Ref 5). SOFE will be used to generate error statistics via Monte Carlo simulation of the linearized first order error equations of the SIGN-III system. A Monte Carlo simulation was chosen instead of a covariance analysis because it was believed that the Monte Carlo approach would require less computer resources (Ref 10).

Three additional supporting computer programs, PROFGEN (Ref 9), SOFEPL (Ref 4), and DISSPLA 9, a utility package of subroutines useful for plotting (REF 13), will be used. PROFGEN is a flight profile generator developed by Musick and SOFEPL, developed by Musick, Feldmann, and Jensen, is a postprocessor for generating sample statistics and plots in conjunction with SOFE. DISSPLA is a post processing/plotting program which will used to produce plots of the SOFEPL results.

While it is difficult to ensure that the error model differential equations are correctly implemented, some degree of confidence in model validity will be achieved by determining the model responses to various initial conditions. The resulting error plots will be compared to those produced in similar fashion by Widnall and Grundy which were published in Reference 6.

A representative C-130 flight profile will be developed and implemented in PROFGEN. The resulting aircraft dynamics data will be used to drive the error differential equations as implented in SOFE. All error sources will be turned on except that no vibration will be used. This will, to some degree, validate that the model has been correctly implemented. Furthermore, this information can be used in a comparison of error magnitudes which can help define the significance of the vibration-induced errors.

The next step will be to develop a means of simulating the vibrational environment. Unfortunately, the normal method of using shaping filters driven by white Gaussian noise to drive the error equations causes an extremely heavy computer burden. The C-130 vibration environment includes frequencies as high as 459 Hz and Shannon's Sampling Theorem would require sampling at a minimum of 918 Hz. For a representative MAC mission of about 8 hours (28,800 seconds), this would require a minimum of 25,920,000 samples. In addition, the vibration power spectral densities consist of very sharp peaks at the C-130 propellor blade passage frequency of 51 Hz and its first 8 harmonics. To generate this power spectral density would require a large bank of shaping filters, which would require a significant amount of computer time per sample. The combination of the two problems makes the normal method of using shaping filters to generate the vibration far too computationally burdensome to be used to inject noise into the system. To alleviate this problem, the vibration will be simulated via two other methods and the results of those methods will be compared.

In the first method, each vibrational peak will be represented by a single sinusoid. All vibrational energy which is outside of these nine peaks will be assumed to contribute little to total system errors and will be ignored. This will result in nine sinusoids for each axis. The resulting sinusoidal series will be used to drive the error differential equations with all other error sources turned off, so that the error due solely to vibration can be determined. A more detailed explanation of this method will be given in Chapter 2.

The second method of simulating the vibration will also represent the vibration as a series of sinusoids, but will assume that sinusoids of different frequencies will combine so that their cotribution will cancel itself out over time. If this assumption is correct, the cross terms cross terms can be ignored. As a result, the number of vibration terms can be reduced by nearly an order of magnitude. This method will be explained in greater detail in Chapter 2.

II. Error Model Development

Basic Error Differential Equations

In order to analyze the performance of an inertial reference system using linear estimation theory, a stochastic system error model may be expressed in the form of a set of linearized stochastic first-order differential equations. These equations are of the form:

$$\dot{\underline{x}} = F\underline{x} + B\underline{u} + \underline{w} \qquad (1)$$

where

F = Fundamental Matrix

B = Control Input Matrix

$\underline{x}$ = Error State Vector

$\underline{u}$ = Deterministic Forcing Function

$\underline{w}$ = White Gaussian Driving Noise

Britting demonstrated that the same set of nine basic equations could be used for both strapdown and gimbaled inertial navigation systems (Ref 1). The set of equations consists of a nine-by-nine matrix system-independent error model (Ref 6) augmented by system-dependent error forcing functions. The first three states represent position errors, the next three are velocity errors, and the last three are tilt errors. Further definition of these states and their units are given in Table 1 on the following page. The matrix for the basic error model (Figure 1) and the corresponding notation (Table 2) are given on the following pages. The first order differential equations for the SIGN-III inertial navigation system used in this study, were developed by Widnall and Grundy (Ref 3).

Coordinate Systems

The basic nine state error model is usually implemented in an east-north-up navigation frame. But, the SIGN-III sensitive axes are aligned in an aircraft body frame oriented down, nose, right-wing (x,y,z). The different coordinate systems makes it necessary to convert sensor noises derived in the body frame into equivalent forces in the navigation frame before using them in the system error model. The orientation of the two frames may be seen in Figure 2 on page 11. The coordinate transformation matrix C_b^n is:

$$C_b^n = \begin{bmatrix} C_{ex} & C_{ey} & C_{ez} \\ C_{nx} & C_{ny} & C_{nz} \\ C_{ux} & C_{uy} & C_{uz} \end{bmatrix} \tag{2}$$

where

$C_{ex} = \sin\psi$

$C_{ey} = -\sin\psi\cos\phi$

$C_{ez} = \cos\psi\cos\phi$

$C_{nx} = \sin\theta \cos\phi$

$C_{ny} = \sin\psi \sin\theta \sin\phi + \cos\phi \cos\theta$

$C_{nz} = \sin\phi \sin\theta \cos\psi - \sin\psi \cos\theta$

$C_{ux} = \cos\psi \cos\theta$

$C_{uy} = \cos\theta \sin\phi \sin\psi - \sin\theta \cos\psi$

$C_{uz} = \cos\theta \sin\phi \cos\psi + \sin\phi \sin\psi$

ϕ = Roll Euler Angle

θ = Pitch Euler Angle

ψ = Yaw Euler Angle

The outputs of the flight profile generator, PROFGEN (Ref 9) are in North-West-Up coordinates. These outputs must be transformed into the body and navigation frames for use in the error differential equations. This is done implicitly in the simulation by equating the west output of PROFGEN to the negative east components in the error model, i.e. one output of PROFGEN is west velocity which is converted via the following equation:

$$v_e = -v_w \tag{3}$$

where

v_e = east velocity in navigation frame

v_w = west velocity in PROFGEN (North-West-Up) coordinates

Once specific force and angular velocity are transformed into the navigation frame, they must also be transformed into the body frame since the sensors are fixed in the body frame.

Table 1. State Definitions for the Fundamental Error Matrix

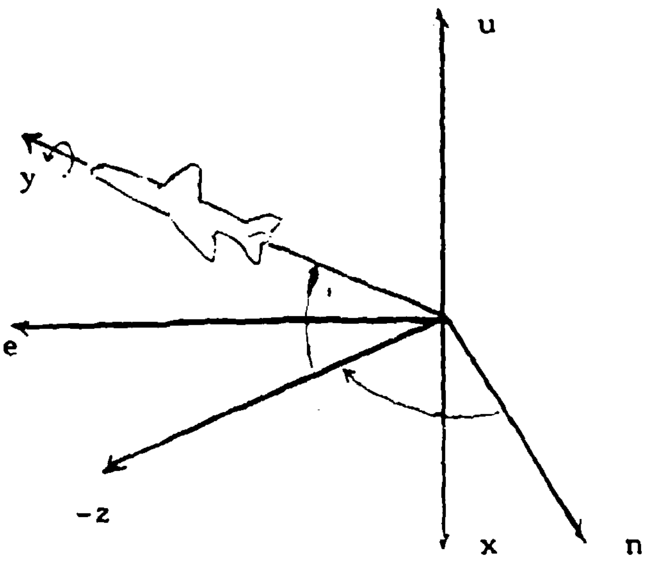
State	Meaning	Units
x(1)	Error in east longitude	radians
x(2)	Error in north latitude	radians
x(3)	Error in altitude	feet
x(4)	Error in east velocity	ft/sec
x(5)	Error in north velocity	ft/sec
x(6)	Error in vertical velocity	ft/sec
x(7)	Error in east attitude	radians
x(8)	Error in north attitude	radians
x(9)	Error in up attitude	feet

$$\begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1 & 0 & \rho_z/\cos L & -\rho_n/R\cos L & 1/R\cos L & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & \rho_e/R & 0 & 1/R & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
4 & 0 & F_{42} & F_{43} & F_{44} & \Omega_z+\omega_z & -\Omega_n-\omega_n & 0 & -f_z & f_n \\
5 & 0 & F_{52} & F_{53} & -2\omega_z & -k_z & \rho_e & f_z & 0 & f_e \\
6 & 0 & -2\Omega_z v_e & F_{63} & 2\omega_n & -2\rho_e & 0 & -f_n & f_e & 0 \\
7 & 0 & 0 & -\rho_e/R & 0 & -1/R & 0 & 0 & \omega_z & -\omega_n \\
8 & 0 & -\Omega_z & -\rho_n/R & 1/R & 0 & 0 & -\omega_z & 0 & \omega_e \\
9 & 0 & F_{92} & -\rho_z/R & (\tan L)/R & 0 & 0 & \omega_n & -\omega_e & 0
\end{array}$$

Figure 1. F Matrix of the INS General Differential Equations

Table 2. Notation used in Figures 1 and 3

Symbol and Value	Meaning
L	Latitude
$\Omega = 7.2921151\times10^{-5}$ rad/sec	Earth rotation rate
$R = 20925640$ ft	Radius of Earth
$g = 32.12698510$	Magnitude of gravity vector
v_e, v_n, v_u	Vel with respect to earth
f_e, f_n, f_u	Components of specific force
$\Omega_n = \Omega \cos L$	North component of earth rate
$\Omega_u = \Omega \sin L$	Up component of earth rate
$\rho_e = -v_n /R$	Components of angular
$\rho_n = v_e/R$	velocity of E-N-Z frame
$\rho_u = (v_e \tan L)/R$	With respect to earth
$\omega_e = \rho_e$	Components of angular
$\omega_n = \rho_n + \Omega_n$	velocity of E-N-Z frame
$\omega_u = \rho_u + \Omega_u$	with respect to inertial space
$k_z = v_u/R$	
$F_{42} = 2(\Omega_n v_n + \Omega_u v_u) + \rho_n v_n/\cos^2 L$	
$F_{43} = \rho_u \rho_e + \rho_z$	
$F_{44} = - k_z - \rho_e \tan L$	
$F_{52} = -2\Omega_n v_e - \rho_n v_e/\cos^2 L$	
$F_{53} = \rho_n \rho_u - k_z \rho_e$	
$F_{63} = 2g/R - (\rho_n^2 + k_z$	
$F_{92} = \omega_n + \rho_u \tan L$	



Note: Origin of body frame is displaced from that of the navigation frame for clarity. They are actually coincident at the aircraft center of mass.

Figure 2. Body and Navigation Coordinate Frames

This is accomplished with a transformation matrix which is the transpose of that in Equation 2.

Altitude Channel Mechanization

The vertical channel of all unaided inertial reference systems are inherently unstable (Ref 6). Typically, a third-order baro-inertial loop is added to stabilze the vertical channnel, but the SIGN-III uses a second-order stabilization loop (Ref 6). The SIGN-III altitude divergence control equation is given in Equation (4):

$$u_{vd} = k_1 \ (h_c - h_{ref}) + k_2 \ (h_c - h_{ref}) \qquad (4) \ \text{(Ref 6)}$$

where

u_{vd} = Baro-inertial aiding variable

h_c = Geocentric altitude above a reference sphere

h_{ref} = Barometric indicated altitude

k_1 and k_2 are the channel gains

The SIGN-III uses the baro-inertial aiding variable to adjust its computed vertical velocity and, thus, stabilizes its vertical channel. However, an error exists in this mechanization, since h_c is a geocentric altitude and h_{ref} is a measure of geodetic altitude. The SIGN-III equations given in Reference 3 do not include any conversion from geodetic to geocentric altitude, and, as a result, the computed geocentric altitude will follow the geodetic barometric altitude. The error is modelled, in Reference 6, as an additional altitude reference error which would be implemented as a deterministic driving term in the error equations (Ref 6). For the purposes of this simulation, it will be assumed that the implemented equations are corrected (or are due only to an error in documentation) by changing the computations so that geodetic altitude is converted to geocentric altitude. Once this assumption is made, the error is eliminated and the following error aiding equations can be written:

$$u_3 = -k_1 (h_c - h_{ref}) \qquad (5)$$

$$u_6 = -k_2 (h_c - h_{ref}) \qquad (6)$$

Equations (5) and (6) are inserted into the differential equations for states x(3) and x(6) respectively. The value of the gains k_1 and k_2 have been selected so that the characteristic equation has a double pole at s = -.01 sec, which provides a loop time constant of about 100 sec. This is the same loop time constant as that used in the Litton Carrier Aircraft Inertial Navigation System or CAINS inertial navigator (Ref 4). The gain values are:

$$k_1 = 3 \times 10^{-2} \text{ sec}^{-1} \qquad (7)$$

$$k_2 = 3 \times 10^{-4} \text{ sec}^{-2} \qquad (8)$$

Gyro Error Model

The SIGN-III uses three torque-to-balance, single-degree-of-freedom floated gas bearing Honeywell GG-334A9 gyros. Since no test data is available on these gyros, Widnall and Grundy based their SIGN-III gyro model on the results of a test of a Honeywell GG-334A16 gyro (Ref 6). This is a low-noise version of the GG-334 series gyro which probably has better performance than the gyros actually used in the SIGN-III. For the purposes of this study, it will be assumed that the test results are representative of GG-334A9 performance.

The gyros are mounted with their input axes nominally orthogonal and are aligned with the platform x, y, z axes (Ref 6). For the purposes of this study, these will be assumed to be nominally aligned with the aircraft body axes. This is a simplification for analysis purposes in order to reduce the computational burden. Typically, a better quality gyro is used for the roll axis, or the gyros are canted relative to the roll axis in order to distribute the roll axis dynamics (Ref 13).

G-insensitive gyro drift, g-sensitive gyro drift, g squared-sensitive gyro drift, gyro scale factor errors, and errors from gyro input axis misalignment will all be achieved in the simulation by augmenting the basic nine state error model with additional states. The g-insensitive gyro drifts will be modeled as random walks, that is:

$$\dot{x} = w \tag{9}$$

The noise, w, will be modeled as being white and Gaussian with strengths as given on the following page. The state initial conditions will be modeled as Gaussian random variables. The standard deviations for the initial conditions are those used in Ref 6 and are listed on the next page.

The remaining gyro effects are of the form:

$$\dot{x} = 0 \tag{10}$$

That is, they are modeled as random biases, and obtained as the output of integrators with zero input, and initial conditions set as above (Ref 6).

One additional simplification has been made in modelling the gyro torquer scale factor errors. The GG-334A9 gyros are torque rebalanced with two torquers per gryo. The primary torquer is used to maintain the gyro orientation, while the secondary torquer is used to compensate for known errors. The primary torquer has a high level mode in which each pulse corresponds to 2×10^{-12} radians and a low level mode in which each pulse corresponds to 2×10^{-15} radians (Ref 6).

The torquer scale factor errors are likely to change depending on which mode is used and/or whether the torquing is in the positive or negative direction (Ref 6). As a result, a complete model of torquer scale factor errors must include 4 states per gyro. This study will use only two torquer scale factor error states per gyro since the C-130 is a large transport aircraft and is unlikely to change its attitude rapidly. As a result, the high level torquing mode will be used rarely, if at all.

Accelerometer Error Model

The SIGN-III uses three single axis rebalancing pendulous accelerometers. They are mounted in a nominally orthogonal configuration along the x, y, and z axis of the platform (Ref 6). The accelerometer biases are modeled as random walks. As was done in Reference 6, the state initial conditions will be set as Gaussian random variables with standard deviations as given in Table 4, as are the strengths of the white Gaussian driving noises.

The SIGN-III error model, from Reference 6, also augments the basic nine-by-nine matrix with additional states for accelerometer scale factor error and accelerometer input axis misalignment. These are modeled as random constants. The initial conditions are modeled as zero-mean Gaussian variables, the standard deviations of which are given on the preceding page.

Gravity Error Model

The SIGN-III gravity model equations are:

$$G_d = -u/r_c^2 \tag{11}$$

$$G_n = 3uJ_2 \sin L_c \cos L_c/r_c^4 \qquad (12)$$

$$G_e = 0 \qquad (13)$$

where

G_d, G_n, G_e = Down, north, east components of gravity

r_c = Geocentric radius

u = Gravitational constant

" = $3.986005 \times 10^{14} m^3/sec^2$

J_2 = Oblateness coefficient of gravitational expansion

L_c = Geocentric latitude

The north equation has an error in it. A correct expression is:

$$G_n = -3uJ_2R^2 \sin L_c \cos L_c/r_c^4 \qquad (13)$$

where R is the equatorial earth radius.

The SIGN-III equation has the wrong sign and the omission of the R^2 term causes the expression to have an extremely small value (Ref 6). Probably, this is an error in the documentation rather than an error in the equations actually used. If these errors are present in the equations used, they would cause an error in north velocity which could be modeled as a deterministic forcing function as follows:

$$u_5 = -0.00162\, g_0 \sin 2L_c \qquad (15)$$

where

u_5 = Non-zero forcing function element

g_o = Equatorial magnitude of gravity for the reference ellipsoid

L_c = Geocentric latitude (Ref 6)

This was done in Widnall and Grundy, but in this simulation, it will be assumed that the correct mechanization is used and no determinsitic driving terms are required.

The error model also includes three states to model the effects of gravity deflections and anomaly. All three states are modeled as first order Gauss-Markov processes with the correlation time being derived from the aircraft velocity and the correlation distance. This is shown in the equation:

$$\dot{x} = -(v/d)\,x + w \tag{16}$$

where

x = Error state

v = Aircraft velocity

d = Correlation distance

w = White noise of strength Q

$Q = 2\sigma^2 v/d$

Reference 6 gives the gravity variation model for the western half of the United States. These figures are shown in the following table:

Table 3. Model for Gravity Variations (Ref 3)

Gravity Error	Standard Deviation	Distance
East-West Deflection	26 g	10 nm
North-South Deflection	17 g	10 nm
Anomaly	35 g	60 nm

Barometric Altimeter Error Model

This simulation will model two sources of error in the barometric altimeter. These errors are scale factor error due to non-standard temperature and the error due to variation in altitude of a constant pressure surface. The non-standard temperature error will be modeled as a random constant scale factor effect with a 0.03 standard deviation. The variation of a constant pressure surface will be modeled as a first order Gauss-Markov process. The correlation distance for the process is 250 nm and the standard deviation is 500 ft (Ref 6).

Truth Model for Zero Vibration

The complete truth model, as developed by Widnall and Grundy in Reference 6, is a 50-state model of the form:

$$\dot{\underline{x}} = F\underline{x} + \underline{u} + \underline{w} \tag{17}$$

The states, x, are defined in Table 5. The fundamental matrix, F, is shown in Figure 3 with its entries being

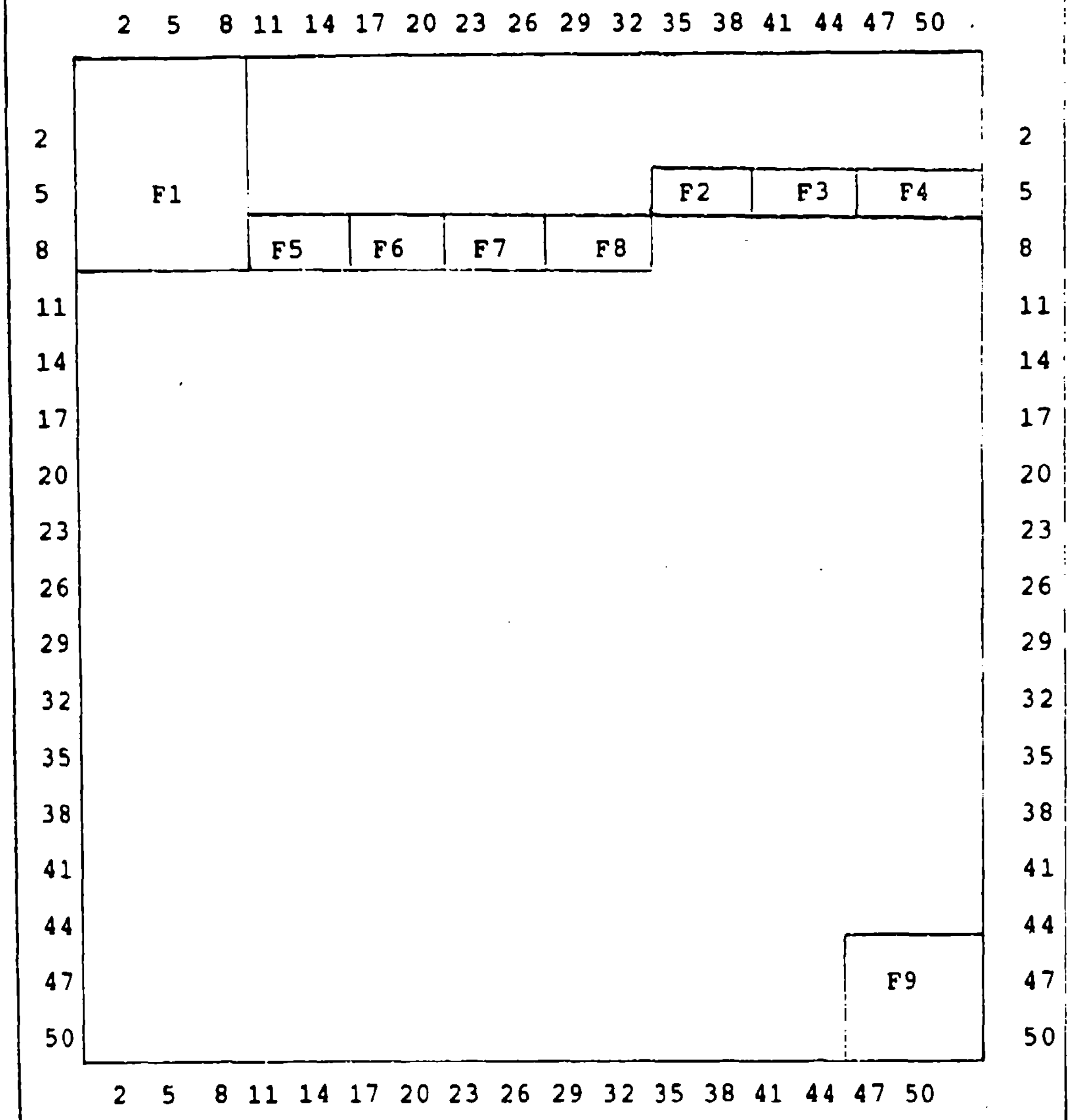


Figure 3a. F Matrix for the 50 State SIGN-III Error Model

$$
\begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1 & 0 & \rho_z/\cos L & -\rho_n/R\cos L & 1/R\cos L & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & \rho_e/R & 0 & 1/R & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
4 & 0 & F_{42} & F_{43} & F_{44} & \Omega_z+\omega_z & -\Omega_n-\omega_n & 0 & -f_z & f_n \\
5 & 0 & F_{52} & F_{53} & -2\omega_z & -k_z & \rho_e & f_z & 0 & f_e \\
6 & 0 & -2\Omega_z v_e & F_{63}-k_1 & 2\omega_n & -2\rho_e & -k_2 & -f_n & f_e & 0 \\
7 & 0 & 0 & -\rho_e/R & 0 & -1/R & 0 & 0 & \omega_z & -\omega_n \\
8 & 0 & -\Omega_z & -\rho_n/R & 1/R & 0 & 0 & -\omega_z & 0 & \omega_e \\
9 & 0 & F_{92} & -\rho_z/R & (\tan L)/R & 0 & 0 & \omega_n & -\omega_e & 0
\end{array}
$$

Figure 3B. Submatrix F1

$$\begin{array}{c|cccccc|c}
 & 34 & 35 & 36 & 37 & 38 & 39 & \\
\hline
4 & C_{ex} & C_{ey} & C_{ez} & C_{ex}f_x & C_{ey}f_y & C_{ez}f_z & 4 \\
5 & C_{nx} & C_{ny} & C_{nz} & C_{nx}f_x & C_{ny}f_y & C_{nz}f_z & 5 \\
6 & C_{zx} & C_{zy} & C_{zz} & C_{zx}f_x & C_{zy}f_y & C_{zz}f_z & 6 \\
\hline
 & 34 & 35 & 36 & 37 & 38 & 39 &
\end{array}$$

Figure 3c. Submatrix F2

$$\begin{array}{c|cccccc|c}
 & 40 & 41 & 42 & 43 & 44 & 45 & \\
\hline
4 & -C_{ex}f_z & C_{ex}f_y & C_{ey}f_z & -C_{ey}f_x & -C_{ez}f_y & C_{ez}f_x & 4 \\
5 & -C_{nx}f_z & C_{nx}f_y & C_{ny}f_z & -C_{ny}f_x & -C_{nz}f_y & C_{nz}f_x & 5 \\
6 & -C_{zx}f_z & C_{zx}f_y & C_{zy}f_z & -C_{zy}f_x & -C_{zz}f_y & C_{zz}f_x & 6 \\
\hline
 & 40 & 41 & 42 & 43 & 44 & 45 &
\end{array}$$

Figure 3d. Submatrix F3

$$\begin{array}{c|ccccc|c}
 & 46 & 47 & 48 & 49 & 50 & \\
\hline
4 & 0 & 0 & 1 & 0 & 0 & 4 \\
5 & 0 & 0 & 0 & 1 & 0 & 5 \\
6 & k_1 - k_2 v/d_1 & k_1 h + k_2 v_z & 0 & 0 & 1 & 6 \\
\hline
 & 46 & 47 & 48 & 49 & 50 &
\end{array}$$

Figure 3e. Submatrix F4

$$\begin{array}{c|cccccc|c}
 & 10 & 11 & 12 & 13 & 14 & 15 & \\
\hline
7 & C_{ex} & C_{ey} & C_{ez} & C_{ex} f_x & C_{ex} f_z & C_{ey} f_y & 7 \\
8 & C_{nx} & C_{ny} & C_{nz} & C_{nx} f_x & C_{nx} f_z & C_{ny} f_y & 8 \\
9 & C_{zx} & C_{zy} & C_{zz} & C_{zx} f_x & C_{zx} f_z & C_{zy} f_y & 9 \\
\hline
 & 10 & 11 & 12 & 13 & 14 & 15 &
\end{array}$$

Figure 3f. Submatrix F5

$$
\begin{array}{c|cccccc|c}
 & 16 & 17 & 18 & 19 & 20 & 21 & \\
\hline
7 & C_{ey} f_z & C_{ez} f_z & -C_{ez} f_y & -C_{ex} f_x f_y & C_{ey} f_x f_z & -C_{ez} f_x f_y & 7 \\
8 & C_{ny} f_z & C_{nz} f_z & -C_{nz} f_y & -C_{nx} f_y f_z & C_{ny} f_x f_z & -C_{nz} f_x f_y & 8 \\
9 & C_{zy} f_z & C_{zz} f_z & -C_{zz} f_y & -C_{zx} f_y f_z & C_{zy} f_x f_z & -C_{zz} f_x f_y & 9 \\
\hline
 & 16 & 17 & 18 & 19 & 20 & 21 &
\end{array}
$$

Figure 3g. Submatrix F6

$$
\begin{array}{c|cccccc|c}
 & 22 & 23 & 24 & 25 & 26 & 27 & \\
\hline
7 & C_{ex} \omega_x^{+} & C_{ex} \omega_x^{-} & C_{ey} \omega_y^{+} & C_{ey} \omega_y^{-} & C_{ez} \omega_z^{+} & C_{ez} \omega_z^{-} & 7 \\
8 & C_{nx} \omega_x^{+} & C_{nx} \omega_x^{-} & C_{ny} \omega_y^{+} & C_{ny} \omega_y^{-} & C_{nz} \omega_z^{+} & C_{nz} \omega_z^{-} & 8 \\
9 & C_{zx} \omega_x^{+} & C_{zx} \omega_x^{-} & C_{zy} \omega_y^{+} & C_{zy} \omega_y^{-} & C_{zz} \omega_z^{+} & C_{zz} \omega_z^{-} & 9 \\
\hline
 & 22 & 23 & 24 & 25 & 26 & 27 &
\end{array}
$$

Figure 3h. Submatrix F7

$$
\begin{array}{c|cccccc|c}
 & 28 & 29 & 30 & 31 & 32 & 33 & \\
\hline
7 & C_{ex}\omega_z & -C_{ex}\omega_y & -C_{ey}\omega_z & C_{ey}\omega_x & C_{ez}\omega_y & -C_{ez}\omega_x & 7 \\
8 & C_{nx}\omega_z & -C_{nx}\omega_y & -C_{ny}\omega_z & C_{ny}\omega_x & C_{nz}\omega_y & -C_{nz}\omega_x & 8 \\
9 & C_{zx}\omega_z & -C_{zx}\omega_y & -C_{zy}\omega_z & C_{zy}\omega_x & C_{zz}\omega_z & -C_{zz}\omega_x & 9 \\
\hline
 & 28 & 29 & 30 & 31 & 32 & 33 &
\end{array}
$$

Figure 3i. Submatrix F8

$$
\begin{array}{c|ccccc|c}
 & 46 & 47 & 48 & 49 & 50 & \\
\hline
46 & -v/d_1 & 0 & 0 & 0 & 0 & 46 \\
47 & 0 & 0 & 0 & 0 & 0 & 47 \\
48 & 0 & 0 & -v/d_2 & 0 & 0 & 48 \\
49 & 0 & 0 & 0 & -v/d_3 & 0 & 49 \\
50 & 0 & 0 & 0 & 0 & -v/d_4 & 50 \\
\hline
 & 46 & 47 & 48 & 49 & 50 &
\end{array}
$$

Figure 3j. Submatrix F9

Table 4. Notation Used in Figure 3

k_1, k_2	Coefficients in altitude channel baro-inertial loop
C_{ij}	Element of the matrix C_b^n, the transformation from SIGN-III coordinates (x,y,z) to local level (east, north, up)
f_x, f_y, f_z	Components of specific force along the SIGN-III x,y,z axes
${}_x, {}_y, {}_z$	Angular velocity of the SIGN-III with respect to inertial space in SIGN-III coordinates
ω_i^+	Equals ω_i if ω_i is positive, else zero
ω_i^-	Equals ω_i if ω_i is negative, else zero
h	Aircraft altitude
v	Aircraft ground speed
d_1	Correlation distance of altimeter error
d_2, d_3, d_4	Correlation distances of gravity deflections and anomaly

Table 5. Error States and Initial Standard Deviations

State	Definition	σ
x(1)	Error in east longitude	5.7735×10^{-2} arc min
x(2)	Error in north latitude	5.7735×10^{-2} arc min
x(3)	Error in altitude	11.45 ft
x(4)	Error in east velocity	1 ft/sec
x(5)	Error in north velocity	1 ft/sec
x(6)	Error in vertical velocity	0.1 ft/sec
x(7)	Error in east attitude	Equation (36)
x(8)	Error in north attitude	Equation (38)
x(9)	Error in up attitude	Equation (40)
x(10)	X gyro drift rate	0.025 deg/hr
x(11)	Y gyro drift rate	0.180 deg/hr
x(12)	Z gyro drift rate	0.180 deg/hr
x(13)	X gyro input axis g-sensitivity	0.2 deg/hr/g
x(14)	X gyro spin axis g-sensitivity	0.2 deg/hr/g
x(15)	Y gyro input axis g-sensitivity	0.2 deg/hr/g
x(16)	Y gyro spin axis g-sensitivity	0.2 deg/hr/g
x(17)	Z gyro input axis g-sensitivity	0.2 deg/hr/g
x(18)	Z gyro spin axis g-sensitivity	0.2 deg/hr/g
x(19)	X gyro input axis g -sensitivity	0.07 deg/hr/g
x(20)	Y gyro spin axis g -sensitivity	0.07 deg/hr/g
x(21)	Z gyro input axis g -sensitivity	0.07 deg/hr/g
x(22)	X gyro pos. scale factor error	70 ppm

Table 5. (Continued)

State	Definition	σ
x(23)	X gyro neg. scale factor error	70 ppm
x(24)	Y gyro pos. scale factor error	70 ppm
x(25)	Y gyro neg. scale factor error	70 ppm
x(26)	Z gyro pos. scale factor error	70 ppm
x(27)	Z gyro neg. scale factor error	70 ppm
x(28)	X gyro input axis misalignment about Y	10 arc sec
x(29)	X gyro input axis misalignment about Z	10 arc sec
x(30)	Y gyro input axis misalignment about X	10 arc sec
x(31)	Y gyro input axis misalignment about Z	10 arc sec
x(32)	Z gyro input axis misalignment about X	10 arc sec
x(33)	Z gyro input axis misalignment about Y	10 arc sec
x(34)	X accelerometer bias	30 μg
x(35)	Y accelerometer bias	20 μg
x(36)	Z accelerometer bias	20 μg
x(37)	X accelerometer scale factor error	35 ppm

Table 5. (Continued)

State	Definition	σ
x(38)	Y accelerometer scale factor error	35 ppm
x(39)	Z accelerometer scale factor error	35 ppm
x(40)	X accelerometer input axis misalignment about Y	10 arc sec
x(41)	X accelerometer input axis misalignment about Z	10 arc sec
x(42)	Y accelerometer input axis misalignment about X	10 arc sec
x(43)	Y accelerometer input axis misalignment about Z	10 arc sec
x(44)	Z accelerometer input axis misalignment about X	10 arc sec
x(45)	Z accelerometer input axis misalignment about Y	10 arc sec
x(46)	Barometric error due to variation in altitude of a constant pressure surface	500 ft
x(47)	Barometric scale factor error	0.03
x(48)	East deflection of gravity	26 μg
x(49)	North deflection of gravity	17 μg
x(50)	Gravity anomaly	35 μg

Table 6. SIGN-III Noise Density Matrix, Non-zero Elements (Ref 6)

Diagonal Element	State Variable	Noise Density
6	Error in vertical Velocity	0.045 v/250 nm
10	X Gyro Drift Rate	$(0.03\ \mathrm{deg/hr})^2/\mathrm{hr}$
11	Y Gyro Drift Rate	$(0.02\ \mathrm{deg/hr})^2/\mathrm{hr}$
12	Z Gyro Drift Rate	$(0.02\ \mathrm{deg/hr})^2/\mathrm{hr}$
34	X Accelerometer Bias	$(10\ g)^2/\mathrm{hr}$
35	Y Accelerometer Bias	$(10\ g)^2/\mathrm{hr}$
36	Z Accelerometer Bias	$(10\ g)^2/\mathrm{hr}$
46	Barometric Pressure	$2\ (500)^2\ v\ /250$ nm
48	East Gravity Deflection	$2\ (26\ g)^2\ v/10$ nm
49	North Gravity Deflection	$2\ (17\ g)^2\ v/10$ nm
50	Gravity Anomily	$2\ (35\ g)^2\ v/60$ nm

Off-Diagonal Element	State Variable	Noise Density
$N_{6,46} = N_{46,6}$	Vertical Velocity	150 v/250 nm

Table 7. SIGN-III Error Source Statistics

State Variable	Noise Spectral Density
	Random Walks
10	$(0.03\ \text{deg/hr})^2/\text{hr}$
11,12	$(0.02\ \text{deg/hr})^2\ /\text{hr}$
34,35,36	$(10\mu g)^2\ /\text{hr}$
	First Order Markov Processes
	Correlation Distance
46	250 n.m.
48	10 n.m.
49	10 n.m.
50	60 n.m.

explained in Tables 2 and 5. The u vector, which is composed of the deterministic driving terms, has only two non-zero elements which are due to mechanization errors. In this simulation, it is being assumed that these errors have been corrected and, as a result, all driving terms will be set to zero. The strengths of the white Gaussian driving terms, w, are given in Table 6. This model will be used as the reference, or control, against which a similar model, modified by the addition of airframe vibration, will be compared.

Vibration Environment

In April of 1982, the 4950th Test Wing, located at Wright-Patterson AFB, Ohio, conducted a vibration flight test of palletized modular electronics rack. The purpose of the test was to establish actual vibration levels at various locations on the pallet during C-130A ground and flight operations (Ref 12). While it is true that this pallet is not identical to those used with operational palletized inertial navigation systems, it was designed for use in the C-130 aircraft and meets military specifications for use with airborne electronics equipment (Ref 12). For the purposes of this study, it will be assumed that the test data contained in Reference 11 is accurate and representative of that to which an inertial navigation system would be subjected when used on board a C-130A aircraft.

The vibration data used in this study are shown in Figures 4, 6, and 8. The energy levels are shown as power

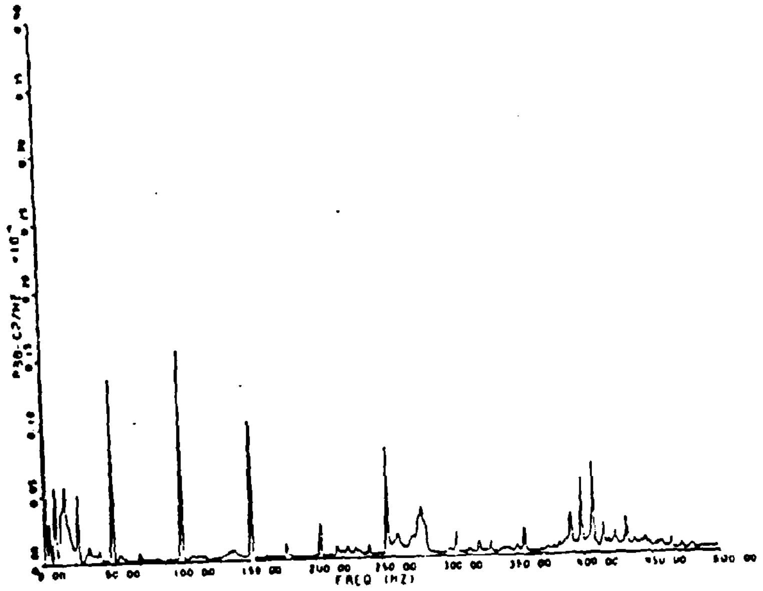


Figure 4. X-Axis Measured Vibration (Ref 12)

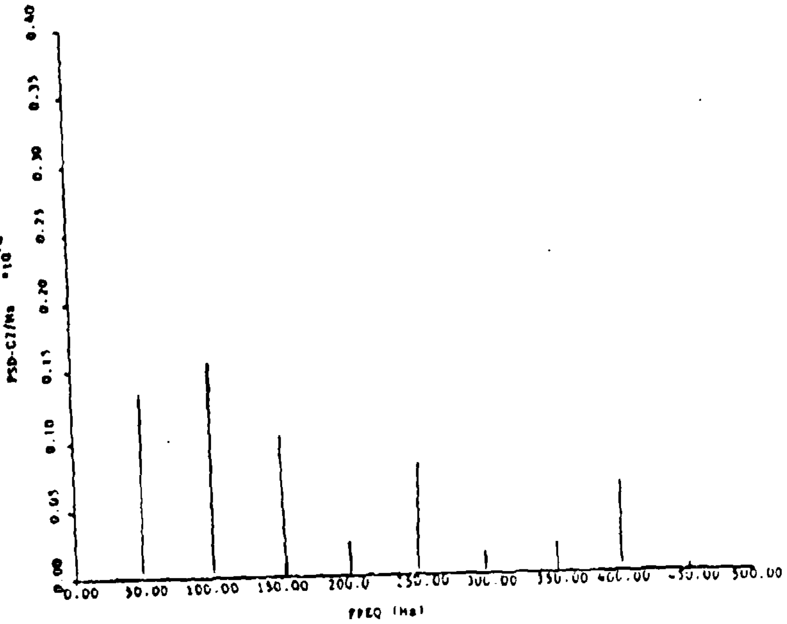


Figure 5. X-Axis Model Vibration

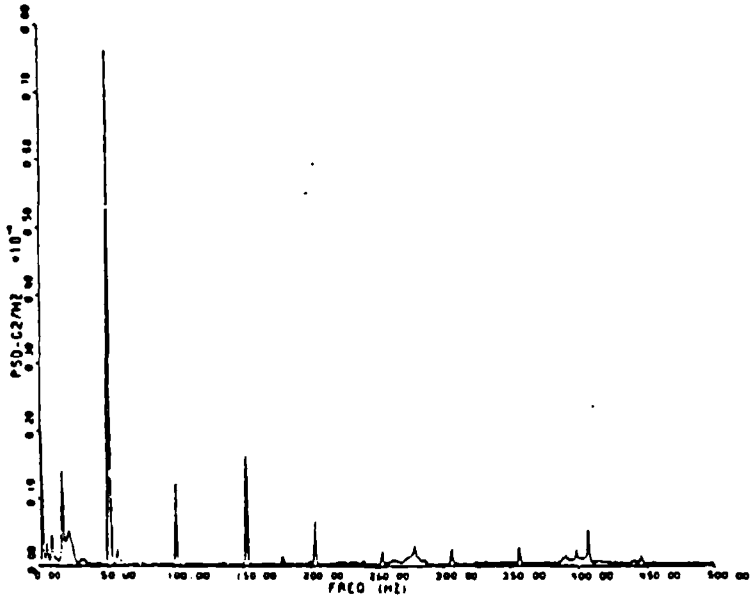


Figure 6. Y-Axis Measured Vibration. (Ref 12)

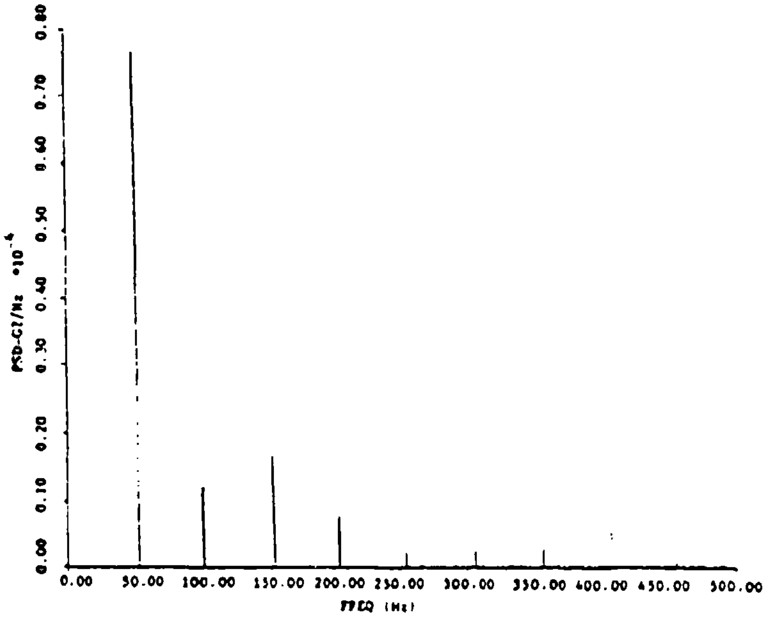


Figure 7. Y-Axis Model Vibration

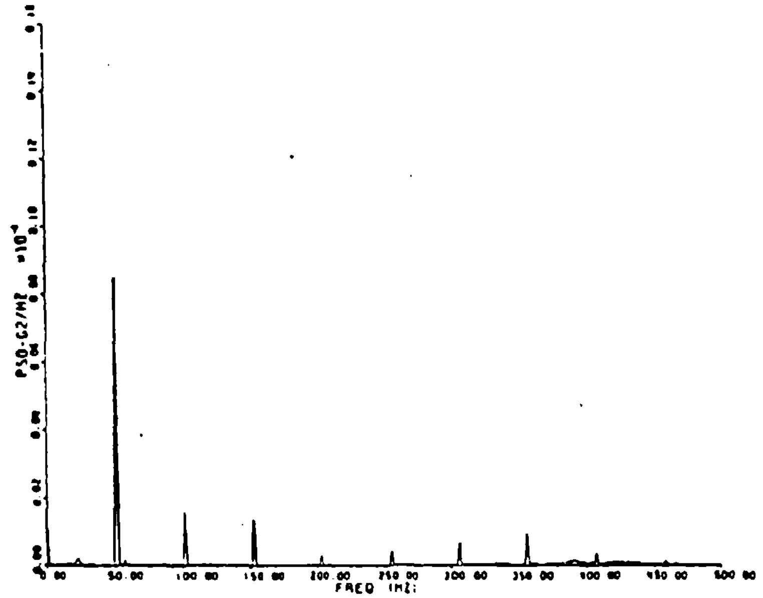


Figure 8. Z-Axis Measured Vibration (Ref 12)

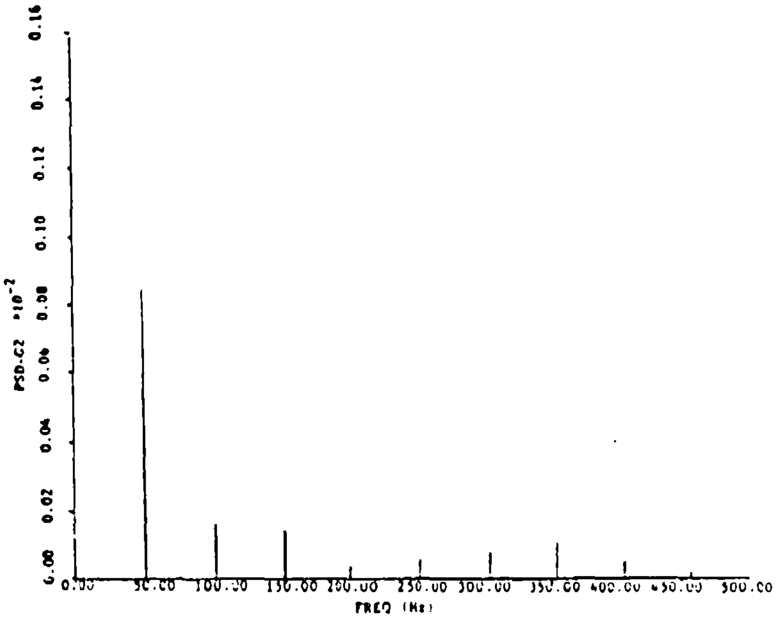


Figure 9. Z-Axis Model Vibration

spectral densities with units of g-squared per Hz. As can be seen in the figures, the majority of vibrational energy occurs at the fundamental and the first eight harmonics of the 51 Hz blade passage frequency. In order to simplify the vibration model, it will be assumed that the vibration outside of these nine peaks contributes little to system errors and will be ignored.

One limitation of the available data is that no information is available about the degree of correlation between the vibration levels in each axis. Although it seems likely that the vibration is correlated to at least some degree, in the absense of any empirical data to the contrary, it will be assumed that the vibration in each axis is uncorrelated with that in the other two axes. This will probably result in some error in the simulation, but it is not anticipated that the error will be excessive.

A good model for vibration is provided by a second order Gauss-Markov process: the output of a second-order system driven by white-Gaussian noise (Ref 8) The general form of the of the output power spectral density of this filter is as follows:

$$\Psi = (a^2\omega^2 + b^2)/(\omega^4 + 2\,\omega_n^2(2\omega^2-1)\,\omega^2 + \omega^4) \qquad (18)$$

This output can be generated by passing a stationary white Gaussian noise of strength Q=1 through a second order system with the following transfer function:

$$G(s) = (as + b)/(s^2 + 2\omega_n s + \omega_n^2) \qquad (19)$$

The system state equations are given below in matrix form.

$$\begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} a \\ c \end{bmatrix} w(t) \qquad (20)$$

The values for a, b, c, ω_n, and ζ are selected to fit the empirical data (Ref 8).

Unfortunately, this method of modeling the vibration requires two additional states for each of the nine peaks in each of the three axis. Thus, to model the vibration via second order Gauss-Markov shaping filters would add 54 additional states to an already large 50-state system model. The sampling rate must also satisfy the Shannon Sampling Theorem, which would require that the sampling rate be at least 918 Hz. As a result, this model of the vibration is not acceptable for use in this study.

In order to reduce the computational burden caused by the high frequency content of the vibration, it was necessary to develop a less computationally demanding method of modelling the vibration. Since the vast majority of the energy in the vibration environment is found in nine peaks, located at the fundamental and the first eight harmonics of the blade passage frequency, it was assumed that the vibration was caused by the rotation of the propellers. Under these conditions, a reasonable model generates the vibration as a series of sinusoids, one sine wave for each of the nine spectral peaks for each axis (Ref 2). If each of the nine sinusoids is then treated as being independent of the other eight, the vibration for each axis can be expressed as:

$$V(t) = A \cos(\omega t + \xi) \qquad \text{(Ref 2, Ref 5)} \quad (21)$$

where

V = the vibration for a given axis

A = square root of twice the magnitude of the corresponding spectral peak

ω = the blade passage frequency

ξ = a randomly generated phase angle.

Figures 4 through 9, which are on the following pages, show the power spectral densities of the vibration levels measured in Ref 12 and those generated by the method described above. It can be seen that the model is not a perfect representation of the actual environment. However, the majority of the energy is contained in the nine spectral peaks and there is no discernible difference between the peaks produced in the actual enviroment and those produced by the model in either frequency or magnitude.

Since it is unlikely that the phase angles will remain constant for the entire flight, the phase angle will be shifted by a randomly generated phase angle every 20 samples. The angle will be generated by the GAUSS subroutine in SOFE and a standard deviation of one degree will be used (Ref 11). Both the rate and standard deviation of the phase shifts were set arbitarily due to a complete absense of data on which to base them.

The computational burden can be further reduced by splitting the model into two parts. One part will include the full fifty-state error model with all driving noises except vibration. This part can be propagated in SOFE with a relatively long sampling period since its associated driving terms do not include any high frequency noises. The second part will include only the vibration terms and those states influenced by the vibration, i.e. the 9 states of the Pinson model and states 13 through 21, a total of 18 states. The vibration enters the model through the g-sensitive gyro drift states (states 13 through 18) and the g-squared sensitive gyro drift states (states 19 through 21). These states, in turn, affect the attitude error states; 7, 8, and 9. The attitude errors are propagated through the normal systems dynamics, and, subsequently, affect the velocity and position error states.

Due to the presence of high frequency noise in the 18-state model, a short sampling period is required. On the other hand, the 50-state model no longer includes any high frequency terms, and the sampling rate may be reduced by several orders of magnitude. A similar study used a sampling period of 2 seconds (Ref 13), and that was used in this case. While reducing the required sampling rate from nearly 1000 Hz to 0.5 Hz (a reduction by a factor of nearly 2000) will not result in a proportionate drop in CPU time, nonetheless, the time required will drop considerably. Since the 50-state simulation does not need to be repeated everytime a new vibration model is simulated, the CPU time required is further reduced. These savings more than compensate for having to simulate 2 separate models.

A further reduction in the computational burden can be made by noting that vibration is zero-mean and symmetrical, and the g-sensitive gyro drift coefficients are constant. As a result, the combination of the vibration and the g-sensitive gyro drift terms will have minimal effect on the system attitude errors. For the purposes of this study, it will be assumed that eliminating the g-sensitive terms from the vibration model will result in negligible errors in the results, and they will not be included in the vibration model. The resulting vibration model consists of the basic nine-state error model with the following terms driving the derivatives of states 7, 8, and 9:

$$V_7 = -C_{ex} f_y f_z Dx - C_{ey} f_x f_z Dy - C_{ez} f_x f_y Dz \tag{22}$$

$$V_8 = -C_{nx} f_y f_z Dx - C_{ny} f_x f_z Dy - C_{nz} f_x f_y Dz \tag{23}$$

$$V_9 = -C_{zx} f_y f_z Dx - C_{zy} f_x f_z Dy - C_{zz} f_x f_y Dz \tag{24}$$

During the remainder of the study, this method will be referred to as the sinusoidal series vibration model. Since both parts of the system error models are linear, the principle of linear superposition allows the following equations to be used to find the error state means and covariances for the system as a whole:

$$E\ (X) = E\ (X1) + E\ (X2) \tag{25}$$

$$E\ (X^2) = (E\ (X1^2) + E\ (X2^2)) \tag{26}$$

This method of modeling the vibration is more efficient than using shaping filters, but it still is not efficient enough since it requires performing 51 additions, 75 multiplications, and evaluating 27 trigonometric functions in order to inject the vibration for each sample and the sampling rate must be fast enough to satisfy Shannon's sampling theorem, or a minimum of 918 samples per second. A more efficient model is required.

Experimental Integrated Effect Vibration Model

The majority of the SIGN-III error model is made up of slowly varing states with only the vibration changing at a rapid rate. For short periods of time, the state transition matrix can be treated as being time invariant. If, during this short period of time, the cumulative effects of the vibration can be determined and simulated, then it is not necessary to generate the vibration; instead the effects can be injected into the rest of the system at periodic intervals.

If the assumptions made in modeling the vibration as a series of sinusoids are valid; then the following equation expresses the relationship between the east attitude error state and the vibration:

$$\begin{aligned} X(t) = &- C_{ex} Dx \sum_{i=1}^{9} (A_i \cos (\omega it + \alpha_i)) \sum_{j=1}^{9} (B_j \cos (\omega jt + \beta_j)) \\ &+ C_{ey} Dy \sum_{k=1}^{9} (C_k \cos (\omega kt + \gamma_k)) \sum_{j=1}^{9} (B_j \cos (\omega jt + \beta_j)) \\ &- C_{ez} Dz \sum_{k=1}^{9} (C_k \cos (\omega kt + \gamma_k)) \sum_{i=1}^{9} (A_i \cos (\omega it + \alpha_i)) \qquad (27) \end{aligned}$$

where

X = east attitude error

Dx, Dx, and Dx = g-square sensitive gyro drift coefficients

A, B, and C = the magnitude of the sinusoids representing the vibration in the x, y, and z axes, respectively.

If, to use as an example, the equation is reduced to show only the effects of vibration in two axes, on the third the equation becomes:

$$X(t) = - C_{ex}Dx \sum_{i=1}^{9} (A_i \cos (\omega i t + \alpha_i))$$

$$\sum_{j=1}^{9} (B_j \cos (\omega j t + \beta_j)) \qquad (28)$$

If this equation is further reduced, again for use as an example, to include only two of the spectral peaks at arbitrary frequencies, ω_1 and ω_2, it becomes:

$$X(t) = - C_{ex}Dx \, (A_1 \cos (\omega_1 t + \alpha_1) + A_2 \cos (\omega_2 t + \alpha_2))$$
$$(B_1 \cos (\omega_1 t + \beta_1) + B_2 \cos (\omega_2 t + \beta_2)) \qquad (29)$$

where

ω_1 and ω_2 = integer multiples of the propellor blade passage frequency

This can be rewritten as:

$$X(t) = - C_{ex}Dx \, (A_1 \, (\cos\omega_1 t \cos\alpha_1 - \sin\omega_1 t \sin\alpha_1)$$
$$+ A_2 \, (\cos\omega_2 t \cos\alpha_2 - \sin\omega_2 t \sin\alpha_2))$$
$$(B_1 \, (\cos\omega_1 t \cos\beta_1 - \sin\omega_1 t \sin\beta_1)$$
$$+ B_2 \, (\cos\omega_2 t \cos\beta_2 - \sin\omega_2 t \sin\beta_2)) \qquad (30)$$

After the terms are multiplied, this becomes:

$$X(t) = -C_{ex}Dx \, (A_1 \, B_1 \, (\cos^2\omega_1 t \cos\alpha_1 \cos\beta_1 - \sin{}_1 t \cos{}_1 t$$

$$(\sin\alpha_1 \cos\beta_1 + \cos\alpha_1 \sin\beta_1) + \sin^2\omega_1 t \sin\alpha_1 \sin\beta_1 \,)$$

$$+ A_1 \, B_2 \, (\cos\omega_1 t \cos\omega_2 t \cos\alpha_1 \cos\beta_2 - \sin\omega_1 t \cos\omega_2 t$$

$$\sin\alpha_1 \cos\beta \; - \sin\omega_2 t \cos\omega_1 t \cos\alpha_1 \sin\beta_2 \; + \sin\omega_1 t$$

$$\begin{aligned}
&\sin\omega_2 t \, \sin\alpha_1 \, \sin\beta_2 \,) + A_2 \, B_1 \, (\cos\omega_2 t \, \cos\omega_1 t \, \cos\alpha_2 \\
&\cos\beta_1 - \cos\omega_1 t \, \sin\omega_2 t \, \sin\alpha_2 \, \cos\beta_1 - \sin\omega_1 t \, \cos\omega_2 t \\
&\cos\alpha_2 \, \sin\beta_1 + \sin\omega_1 t \, \sin\omega_2 t \, \sin\alpha_2 \, \sin\beta_1 \,) \\
&+ A_2 \, B_2 \, (\cos^2\omega_2 t \, \cos\alpha_2 \, \cos\beta_2 \\
&- \sin\omega_2 t \, \cos\omega_2 t \, (\sin\alpha_2 \, \cos\beta_2 + \cos\alpha_2 \, \sin\beta_2 \,) \\
&+ \sin^2\omega_2 t \, \sin\alpha_2 \, \sin\beta_2 \,)) \qquad (31)
\end{aligned}$$

This expression can be solved to find X(T) by integrating both sides of the equation. If the limits of integration are selected so that the time interval is an integer multiple of pi divided by 51 (the blade passage frequency), X(T) is as follows:

$$\begin{aligned}
X(T) = (-C_{ex}Dx \; T/2) \; (&A_1 \, B_1 \, (\cos\alpha_1 \, \cos\beta_1 + \sin\alpha_1 \, \sin\beta_1 \,) \\
&+ A_2 \, B_2 \, (\cos\alpha_2 \, \cos\beta_2 + \sin\alpha_2 \, \sin\beta_2 \,)) \qquad (32)
\end{aligned}$$

where T = the time interval

By using trigonometric identities, this can be expressed as:

$$\begin{aligned}
X(T) = (-C_{ex}Dx \; T/2) \; (&A_1 \, B_1 \, (\cos(\alpha_1 - \beta_1) + \cos(\alpha_1 + \beta_1) \\
&+ \cos(\alpha_1 - \beta_1) - \cos(\alpha_1 + \beta_1)) \\
&+ A_2 \, B_2 \, (\cos(\alpha_2 - \beta_2) + \cos(\alpha_2 + \beta_2) + \cos(\alpha_2 - \beta_2) \\
&- \cos(\alpha_2 + \beta_2))) \qquad (33)
\end{aligned}$$

This, in turn, reduces to:

$$X(T) = -C_{ex}Dx\ T\ (A_1\ B_1 \cos\ (\alpha_1-\beta_1) + A_2\ B_2 \cos\ (\alpha_2-\beta_2)) \quad (34)$$

Thus, it can be seen that if T is selected small enough that the state transition matrix can be assumed constant over the time interval, and if T is selected to be an integer multiple of pi divided by the blade passage frequency, then this method of simulating the effects of the vibration should produce results of acceptable accuracy. In order to check the accuracy of this method, short Monte Carlo simulations of each method will be accomplished. The results of these runs will be compared in Chapter IV.

Initial Conditions

The initial conditions for the first nine states will be based upon an assumed ground alignment at a random heading. It will also be assumed that the baro-inertial vertical channel has reached a steady state condition.

The initial longitude and latitude errors, $x_o(1)$ and $x_o(2)$, will be based on the assumption that the navigator can enter the alignment position to within an error of +/- 0.1 arcminute. Since the variance of a variable, distributed uniformly over a range T, is T /12, the square root of the variance (or standard deviation) may be calculated with the following equation:

$$\sigma = (0.2 \text{ arc min})/\sqrt{12} = 5.7735 \times 10^{-2} \text{ arc min (Ref 13)} \quad (35)$$

During the Monte Carlo simulation, the initial conditions of x(1) and x(2) will be generated by a zero-mean Gaussian distributed variable based on the standard deviation given in Equation (34). A Gaussian distribution will be used instead of a uniform distribution since a means of generating a Gaussian sample is already available in SOFE and the resulting error should be minimal.

The initial condition for the altitude error, x(3) , will be generated based upon the assumption that the aircrew will be able to enter the altitude to within 20 feet. This will be approximated by a Gaussian distributed variable with a standard deviation of $2(20)/\sqrt{12}$ or 11.45 feet (Ref 13).

The initial standard deviations of the velocity states, x(4), x(5), and x(6), will be those recommended in Reference 6. These are:

$$\sigma(4) = 1.0 \text{ ft/sec}$$

$$\sigma(5) = 1.0 \text{ ft/sec}$$

$$\sigma(6) = 0.1 \text{ ft/sec}$$

The initial condition for the vertical velocity, X(6), is more accurate than the other velocity initial conditions due to barometric altimeter aiding.

The east and north attitude errors, x(7) and x(8) , depend on the initial accelerometer and gravity errors. In the alignment process, the transformation matrix from the platform reference frame to the navigation frame is rotated into alignment with the sensed gravity vector. This causes the initial attitude errors to correspond to errors in the sensed gravity vector. The alignment heading is a factor in the contribution of each sensor to the initial state errors (Ref 13).

Row five of the fundamental matrix can be used to show the relationship between the initial east tilt error, x (7) , and acceleration errors (Ref 13). For an aircraft at rest, the equation is:

$$0 = -2\Omega \sin L\ x_o(4) + x_o(7)\ f_u + x_o(34)\ C_{nx} - x_o(35)\ C_{ny} - x_o(43)\ C_{ny}\ f_u - x_o(49) \qquad (36)$$

where

Ω = magnitude of the earth's angular velocity

f_u = force up in the body frame

C_{nx} , C_{ny} , C_{nz} = elements of the body frame to navigation frame transformation matrix

But for an aircraft at rest, $f_u = -f_z = g$ (where f_z is the force in the z direction of the navigation frame), and assuming alignment at a random heading , the equation becomes:

$$x_o(7) = 2\Omega \sin L\ x_o(4)/g + \cos\ x_o(34)\ -\sin\ (x_o(35)/g - x_o(43)) - x_o(49)/g \qquad (37)$$

The fourth row of the fundamental matrix can be used to find the relationship for the initial north tilt error $x_o(8)$ (Ref 13)

$$\omega_x = -\Omega_n \cos\Psi + \Omega_u \sin\Psi$$

$$\omega_y = \Omega_u \cos\Psi + \Omega_u \sin\Psi$$

Flight Profile

The object of the flight profile is to excite the long term error modes of the system. No attempt was made to follow a specific mission from a given base along a given flight path, but rather a profile was developed which included representative mission segments for a C-130 aircraft.

Mission Profile

The mission profile simulates a generic C-130 mission. The flight includes takeoff, a simple departure, a high level cruise leg, and a short low level route. The 20 segments for this mission are listed in Table 7 on the next page. In the table, time is given in seconds duration for the segment. The acceleration vector is divided into two components; one along the route of flight, and the other tangential to the flight path as generated by the aircraft maneuvers.

The mission starts lined up on the runway with zero velocity. The start point is 35 degrees north latitude and 90 degrees west longitude at sea level. The runway heading is an arbitrary 315 degrees. From this initial condition, the aircraft accelerates down the runway until it reaches a ground speed of 105 knots. The wind is assumed to be zero. At this point, the aircraft pitches up 3 degrees and starts to climb. During the climb, it continues to accelerate to a climb airspeed of 180 knots. The departure includes two heading changes; the first is 90 degrees right followed by a second, at level-off, of 45 degrees right. The entire departure takes just under 33 minutes. At level-off, the heading is straight east (heading 090 true), and the aircraft begins accelerating to a cruise airspeed of 280 knots. It should be noted that the ground speed is also 280 knots since the wind is assumed to be calm. The high altitude cruise leg is exactly 6 hours long.

At the end of the high altitude cruise leg, the aircraft makes a 90 degree right turn to a heading of 180 degrees and starts to slow to its low altitude cruise airspeed of 260 knots. At the descent point, the aircraft pitches down 5.1 degrees and descends at 260 knots. The descent takes 11 minutes and continues until the aircraft levels off at about 900 feet. The low level portion of the mission has 10 segments, each 10 minutes long. The low level segments are separated by turns of 45, 90, or 135 degrees. The mission ends at low level after a total flight time of 9 hours 2 1/2 minutes.

Summary

Equation 1 expressed the form of a set of linearized, stochastic, first-order differential equations which can be used to represent an error model of the SIGN-III inertial navigation system. Equation 1 was modified by the addition of two different models of the vibration environment of a

C-130A, as measured during flight tests. The vibration models represent the enviroment through a series of sinusoids. While an accurate representation is crucial, of equal importance is the requirement that the model be computationally efficient since the vibration to be modeled includes frequencies as high as 459 Hz. In addition, flight mission data is necessary to provide inputs to the system model.

Table 8. Flight Profile

Segment	Time	Maneuver	Degrees	Tangent g's	Path g's
1	28.5	Straight	0	0	0.195
2	202	Pitch	3	0.1	0.0196
3	605	Turn	90	0.5	0
4	875	Turn	45	0.5	0
5	260	Pitch	-3	0.5	0.0194
6	19800	Straight	0	0	0
7	400	Turn	90	0.5	0
8	60	Straight	0	0	-0.01885
9	660	Pitch	-5.1	0.25	0
10	60	Pitch	5.1	0.25	0
11	600	Turn	-45	0.5	0
12	600	Turn	90	0.5	0
13	600	Turn	-135	0.5	0
14	600	Turn	90	0.5	0
15	600	Turn	-135	0.5	0
16	600	Turn	-45	0.5	0
17	600	Turn	45	0.5	0
18	600	Turn	-90	0.5	0
19	600	Turn	45	0.5	0
20	600	Turn	-90	0.5	0

III. Software

The primary program used in this study is a Monte Carlo simulation program developed by the Air Force Avionics Laboratory called SOFE (Ref 10). Two additional programs used with SOFE are a flight profile generator, PROFGEN (Ref 9), and a statistical/plotting postprocessor for SOFE, SOFEPL (Ref 4).

SOFE

This section will discuss the implementation of SOFE for the system simulation. Appendix A has a discussion of the program itself. SOFE can be used to implement both a complete truth model and a reduced order Kalman filter model. However, in this simulation Kalman filter performance was not an issue and the filter was essentially eliminated by making its order one, the minimum allowable in SOFE.

In a Monte Carlo simulation, the truth model is propagated through many runs. After the set of runs is completed, the accumulated data can be analyzed to determine the sample mean and variance of the states. These statistical computations are done by SOFEPL.

SOFEPL

SOFEPL is a postprocessor for SOFE. It is capable of performing statistical computations such as generating ensemble averages and standard deviations for error states. It uses a graphics package, DISSPLA, to create a plot file. The DISSPLA postprocessor can be used to make the actual plots.

PROFGEN

Although trajectory information can be generated in SOFE, a separate program, PROFGEN, was used to simplify programming and to reduce the amount of computer resources required for any given run. A description of PROFGEN is given in Appendix B.

PROFGEN can be asked to generate a total of twenty output variables, but only seventeen were used in this simulation. The meanings and units for the output variables used are given on the next page. Many of the output variables required transformation into different coordinate frames before they could be used in SOFE. PROFGEN uses a north-west-up coordinate frame and the SOFE simulation was run in an east-north-up navigation frame and a fuselage bottom-nose-right wing body frame. The transformation from the PROFGEN frame to the navigation frame was easily handled by equating the corresponding components of each vector.

Specific force, now in the navigation frame, had to be transformed into the body frame using C_n^b from Equation (2) as shown:

Table 9. PROFGEN Output

Symbol	Definition	Units
t	Time	seconds
L	Latitude	radians
l	Longitude	radians
h	Altitude	feet
α	Heading	radians
ϕ	Roll	radians
θ	Pitch	radians
ψ	Yaw	radians
$\dot{\phi}$	Roll Rate	rad/sec
$\dot{\theta}$	Pitch Rate	rad/sec
$\dot{\psi}$	Yaw Rate	rad/sec
v_n	North Velocity in Nav Frame	ft/sec
$-v_e$	West Velocity in Nav Frame	ft/sec
v_u	Up Velocity in Nav Frame	ft/sec
f_n	Specific Force Along North Axis	ft/sec
$-f_e$	Specific Force Along East Axis	ft/sec
f_u	Specific Force Along Up Axis	ft/sec

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} C_{xe} & C_{xn} & C_{xu} \\ C_{ye} & C_{yn} & C_{yu} \\ C_{ze} & C_{zn} & C_{zu} \end{bmatrix} \begin{bmatrix} f_e \\ f_n \\ f_u \end{bmatrix} \tag{42}$$

The angular velocity of the body frame with respect to inertial space must also be computed from the PROFGEN outputs. It can be found as the sum of the angular velocity of the earth with respect to inertial space plus the angular velocity of the body frame with respect to the earth. The results of these computations in the body frame are

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} C_{ye} & C_{ye} & C_{ye} \\ C_{ye} & C_{ye} & C_{ye} \\ C_{ye} & C_{ye} & C_{ye} \end{bmatrix} \begin{bmatrix} \omega_e \\ \omega_n \\ \omega_z \end{bmatrix} + \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{43}$$

where

$$\omega_e = -v_n/R$$
$$\omega_n = v_e/R + \cos L$$
$$\omega_u = (v_e \tan L)/R + \Omega \sin L$$

$$R = 20925640 \text{ ft}$$

$$\Omega = 7.2921151 \times 10^{-5} \text{ rad/sec}$$

Summary

This study used three software programs: SOFE, SOFEPL, and PROFGEN. SOFE is a Monte Carlo simulation program which was used to propagate the SIGN-III state equations. A flight profile generator, PROFGEN, was used to provide data about the aircraft dynamics to SOFE. SOFEPL was used as a statistical/plotting postprocessor for the SOFE outputs.

IV. Simulation Results

Program Validation

The implementation of the basic nine-by-nine error matrix will be validated by comparing its response to various initial conditions to those obtained by Widnall and Grundy (Ref 3). They generated plots showing the response of the unstable, unaided basic nine-by-nine system model to initial errors in position, velocity, and attitude. The plots in Reference 3 include the response to initial errors in latitude of one arc minute, an altitude error of ten feet, north, east and up velocity errors of one foot per second, and north, east and up attitude errors of one arc minute. These plots were duplicated, but only a limited number are included for sake of brevity. Figures 10 and 11 show the effect of an initial latitude error on north attitude and latitude. Figures 12 and 13 are plots of the longitude and east attitude errors resulting from initial errors in north and up attitudes, respectively. The errors induced in latitude and east velocity by a one foot per second error in east velocity are shown in Figures 14 and 15.

A small scale was used in Reference 12, and, in order to allow for direct comparison, during this study. Due to the small scale, it is difficult to be certain that the plots agree completely. However, there is no discernable difference in the plots.

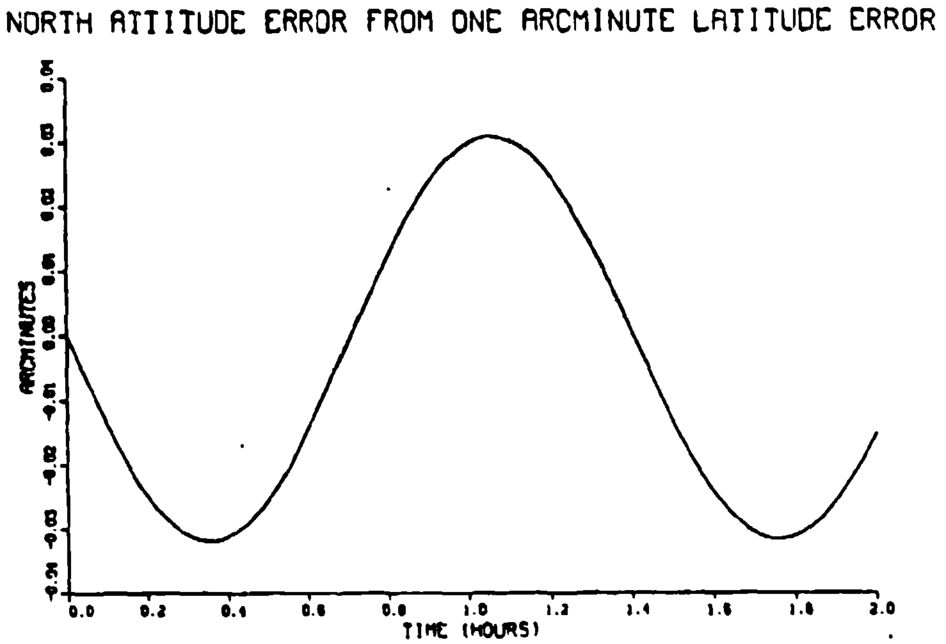


Figure 10. N. Attitude Error for I.C. of x(2)=1 arcmin

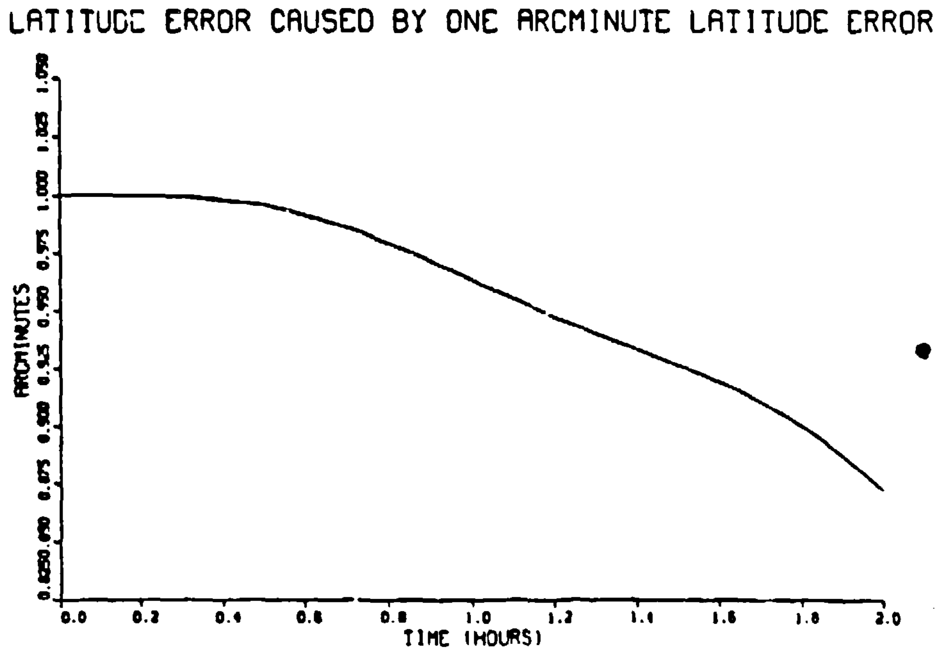


Figure 11. Latitude Error for I.C. of x(2)=1 arcmin

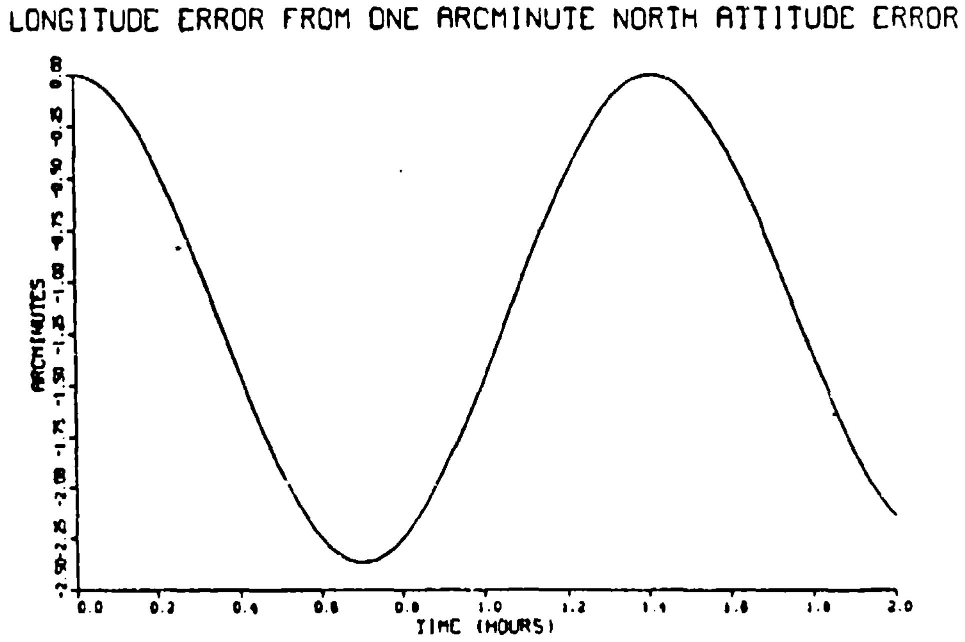


Figure 12. Longitude Error for I.C. of x(3)=1 arcmin

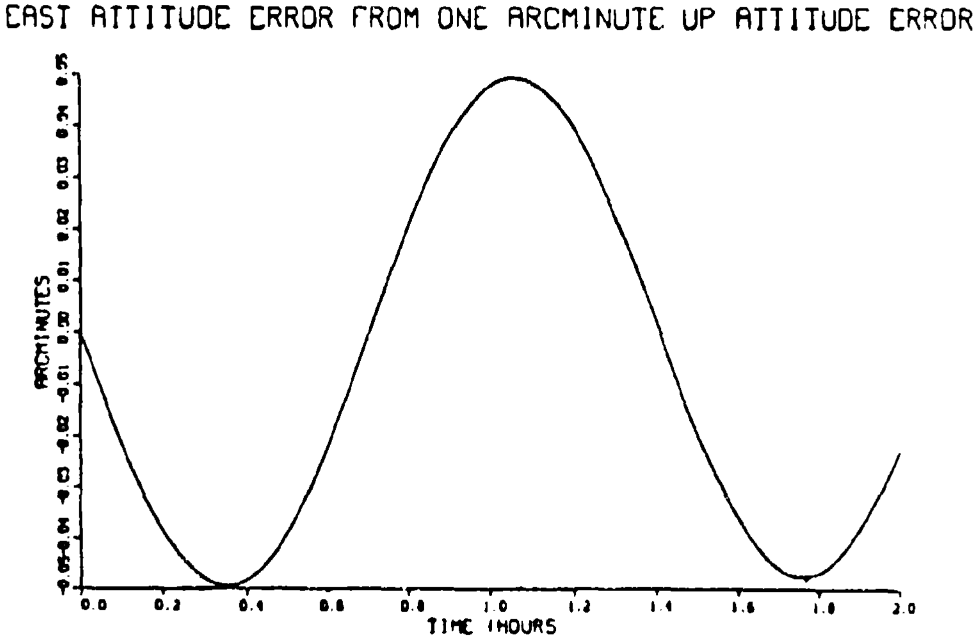


Figure 13. East Attitude Error for I.C. of x(8)=1 arcmin

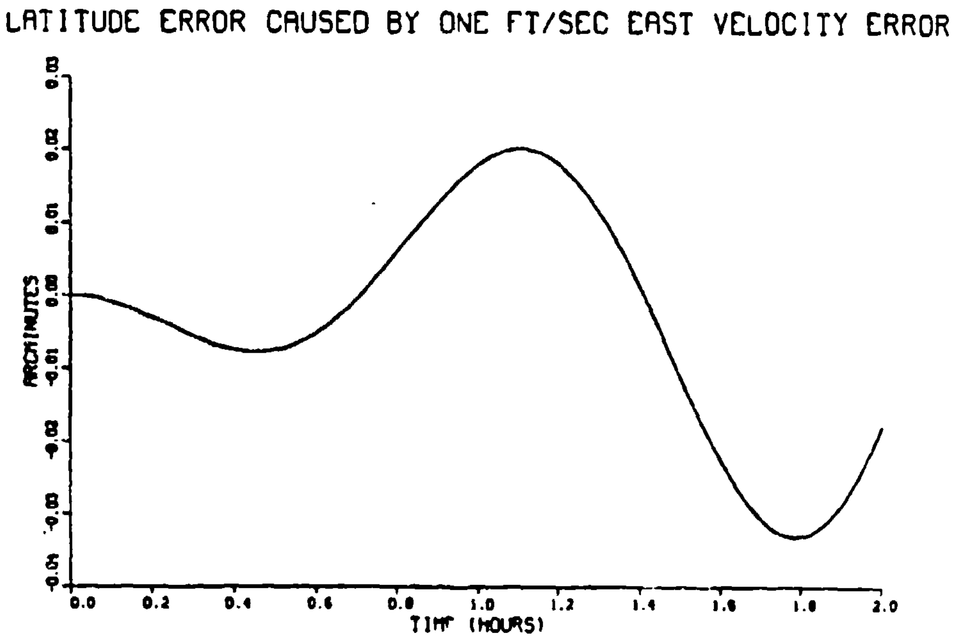


Figure 14. Latitude error for I.C. of x(4)=1 ft/sec

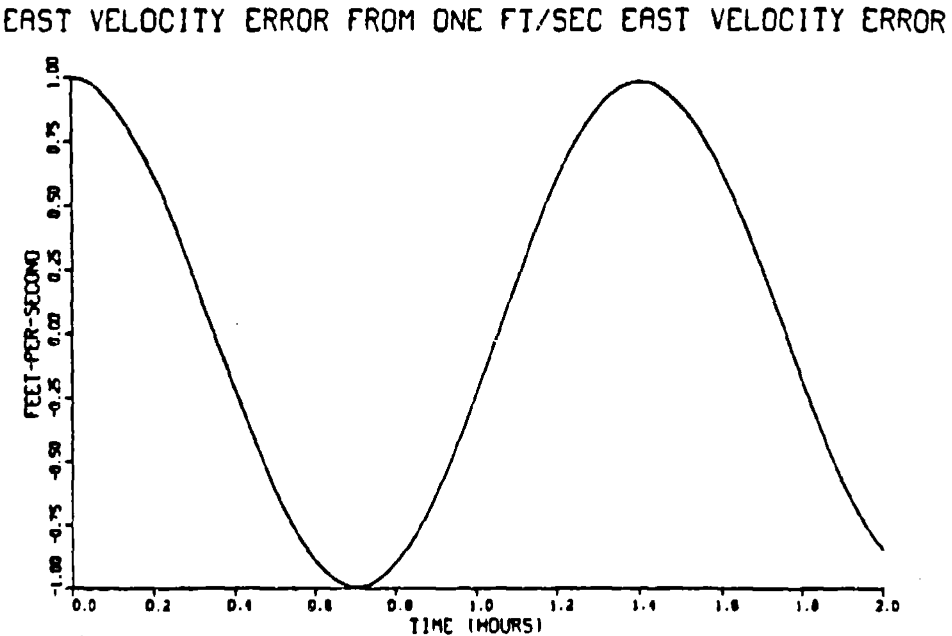


Figure 15. East Velocity Error for I.C. of x(4)=1 ft/sec

Chapter 2 included the development of the basic error model and two different methods of simulating the C-130A vibration environment. In this section, the results of Monte Carlo simulation using those models will be presented. The two different methods of modeling the vibration, the sinusoidal series vibration model and the experimental integrated effect model, were simulated in separate 15-minute, 20-run Monte Carlo simulations. During these simulations, the only aircraft motion was that simulated by the vibration models. The results of these runs are shown in Figures 16 through 25. Subsequently, the second simplified method was simulated for an 8-hour mission with a 20-run Monte Carlo simulation. The results of these runs are compared with a 20-run simulation of the 50-state basic error model without vibration in Figures 26 through 37.

Vibration-Induced Errors

While 20 runs of a 15-minute mission do not provide a complete picture of the vibration-induced errors, computer limitations prevented the use of a longer simulation. The 20-run simulation of the sinusoidal series model required nearly 4 hours of cpu time on a CDC Cyber 74 computer. The experimental integrated effect model required significantly less time, but still used approximately one hour of cpu time.

Over the 15-minute runs, the vibration-induced errors, in both cases, were very small. After 15 minutes, the standard deviations of the latitude error for the first

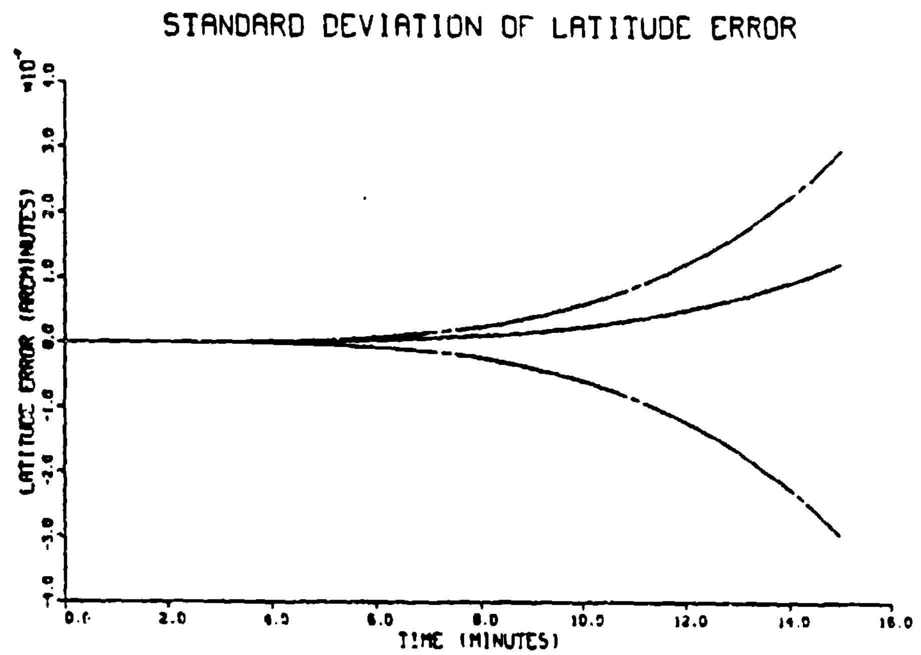


Figure 16. Latitude Error for Sinusoidal Method

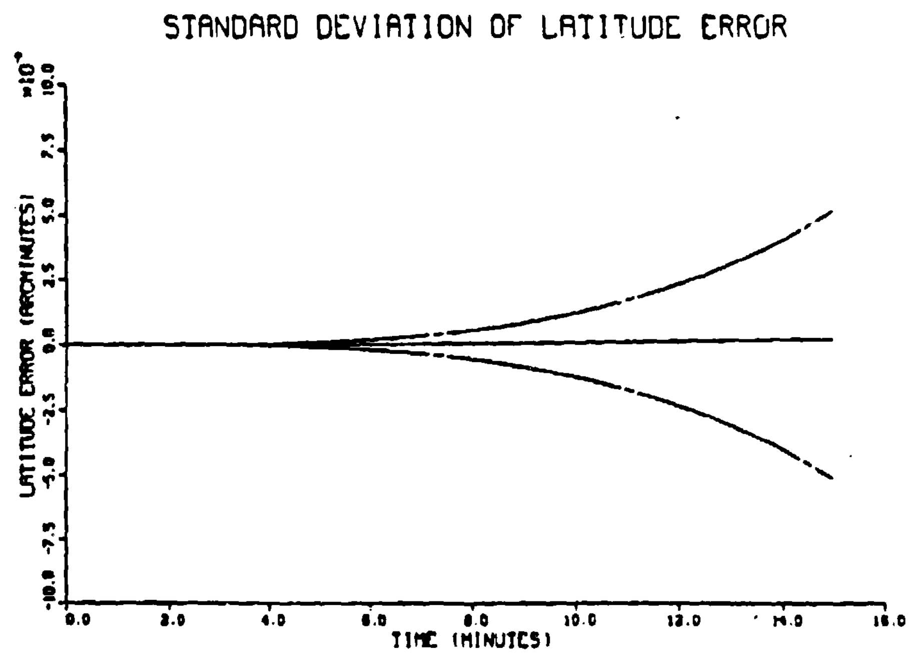


Figure 17. Latitude Error for Integrated Effect Model

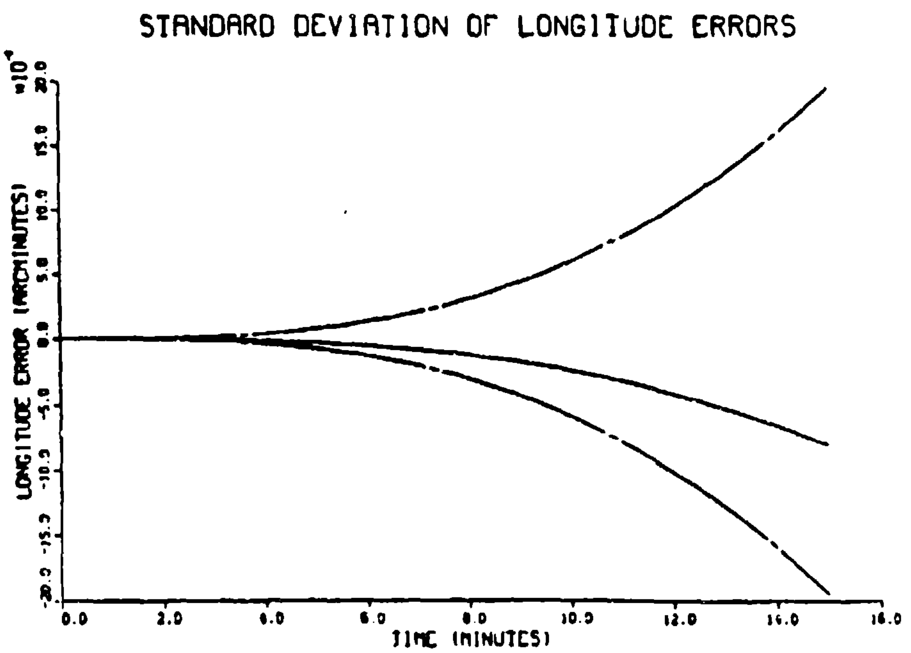


Figure 18. Longitude Error for Sinusoidal Method

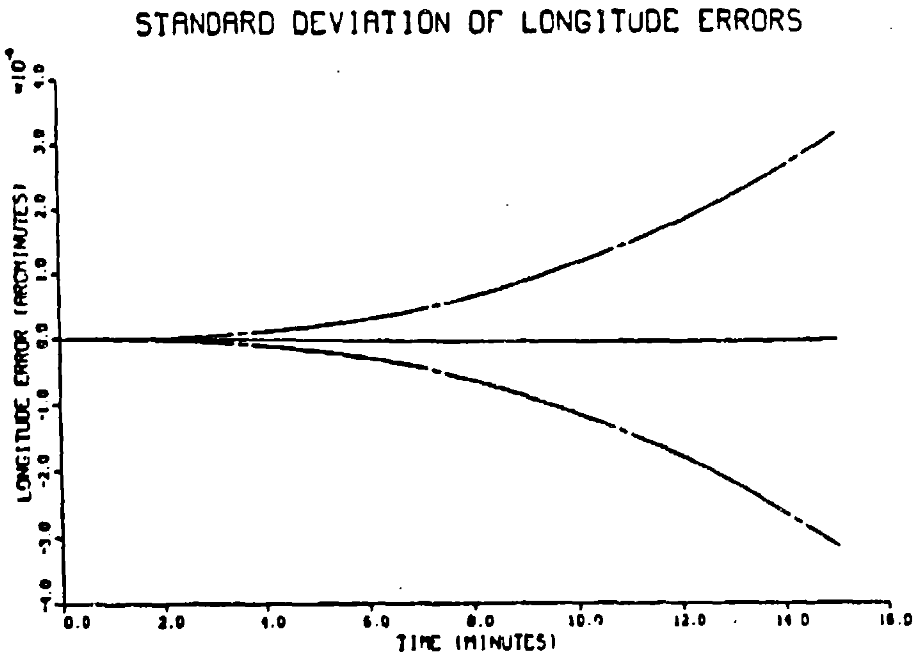


Figure 19. Longitude Error for Integrated Effect Method

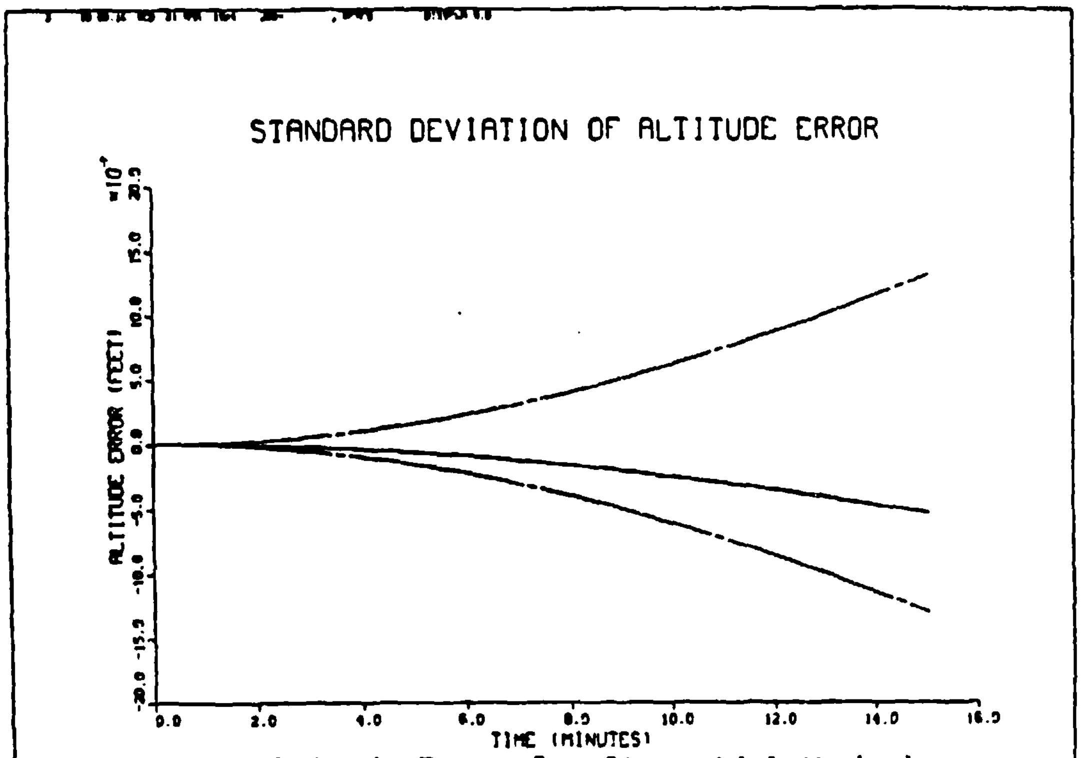


Figure 20. Altitude Error for Sinusoidal Method

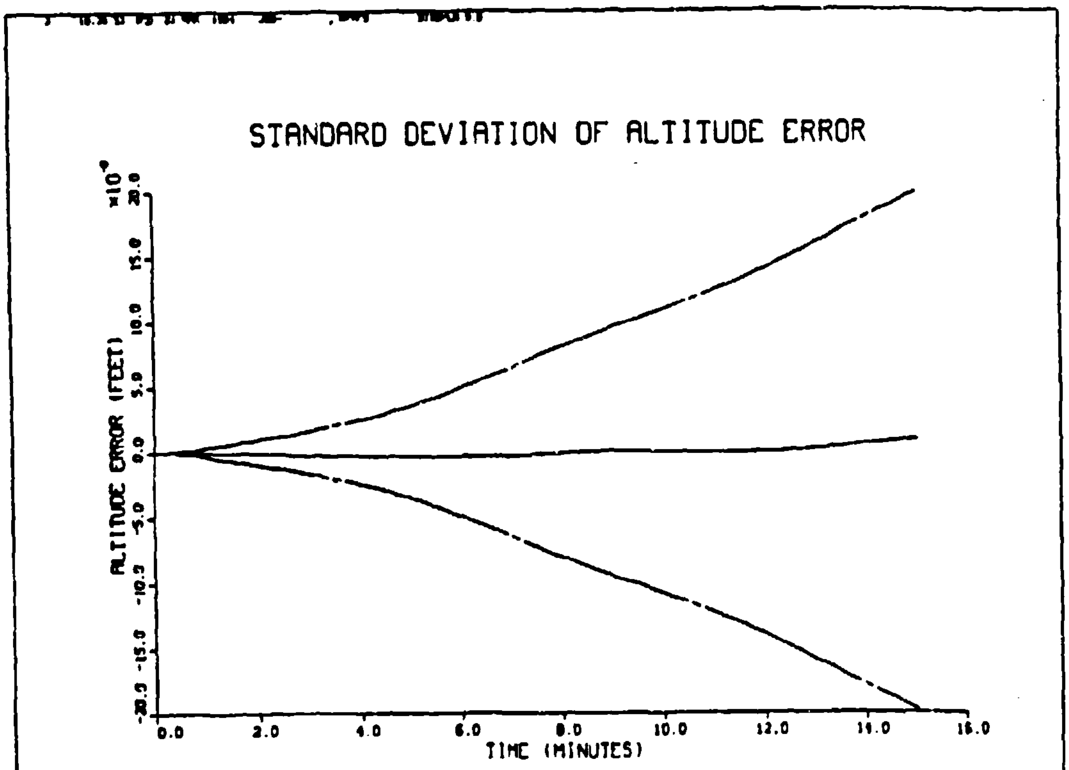


Figure 21. Altitude Error for Integrated Effect Method

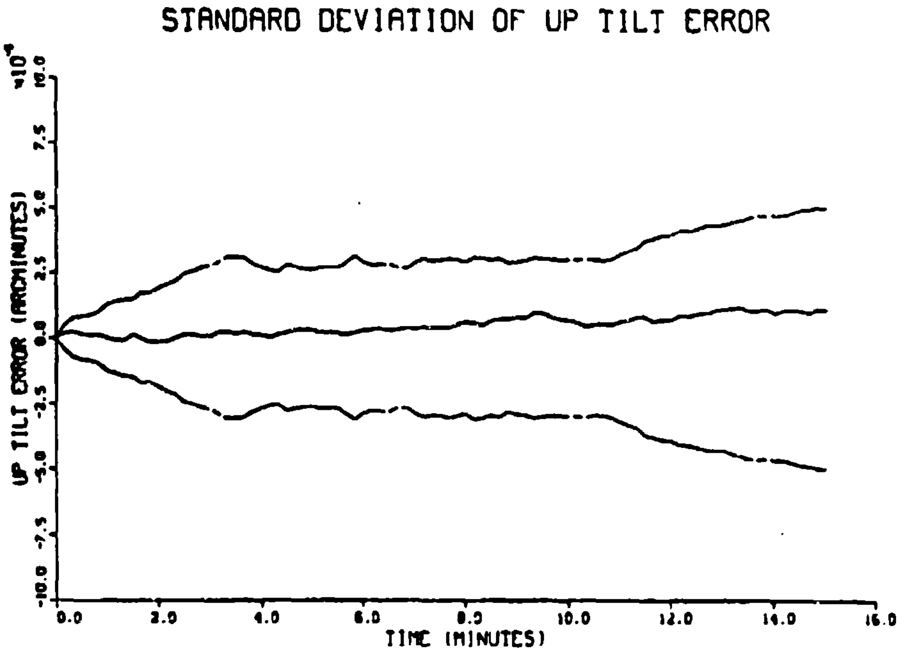


Figure 22. Up Tilt Error for Sinusoidal Method

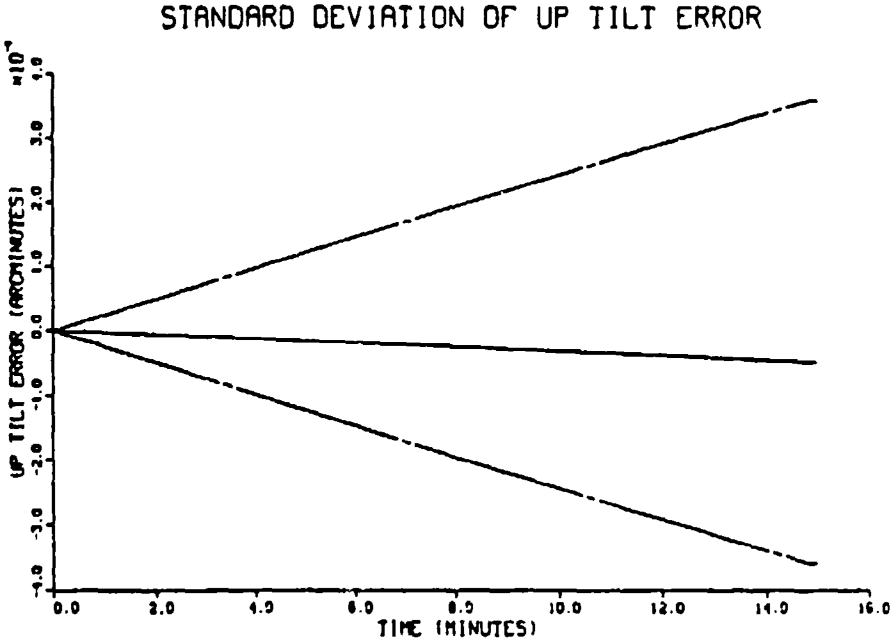


Figure 23. Up Tilt Error for Integrated Effect Method

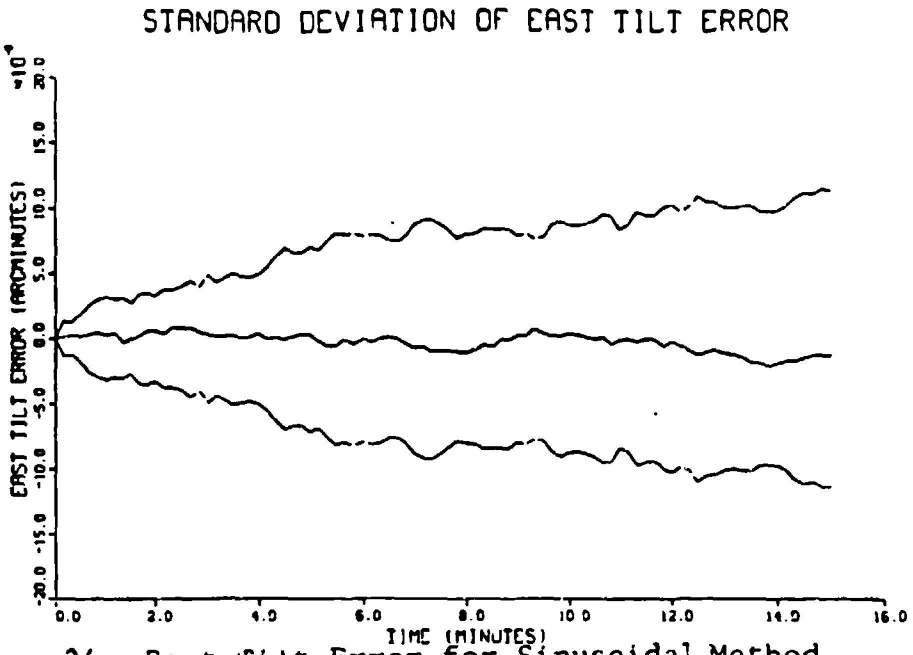


Figure 24. East Tilt Error for Sinusoidal Method

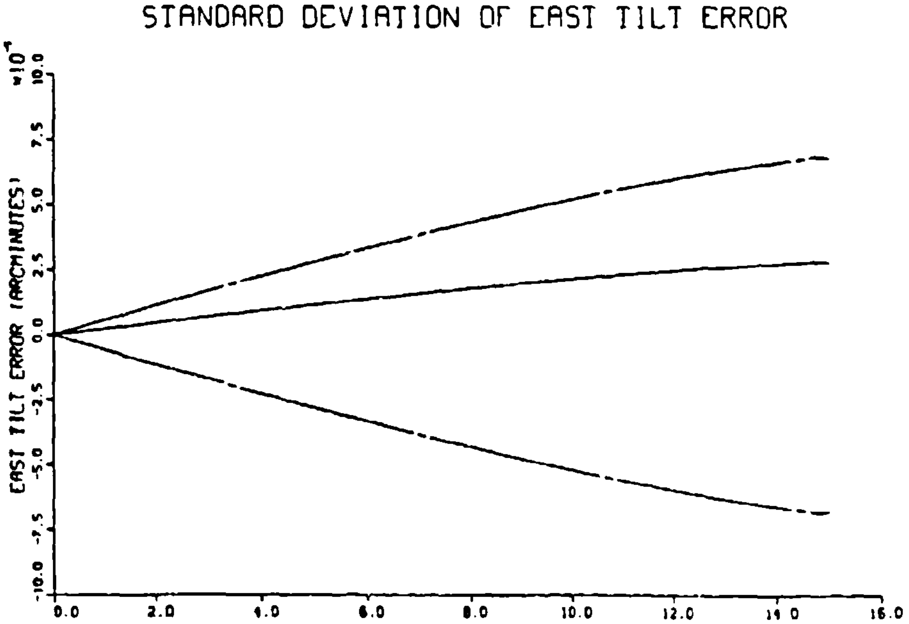


Figure 25 East Tilt Error for Integrated Effect Method

method were approximately 0.3 micro-arcminutes or about 0.026 inches. The standard deviation of the latitude error for the second method was even smaller, about 0.005 inches. The error standard deviations from the second simulation were smaller, by a factor of about 4. The longitude errors were larger; from the first simulation, the standard deviation of longitude errors after 15 minutes was about 20 micro-arcminutes (1.5 inches). For the second method, the errors were again smaller; the standard deviations were about 3.5 micro-arcminutes (about 0.25 inches).

The plots for latitude, longitude, and altitude demonstrate another significant difference in the results produced by the two methods. The plots for the sinusoidal model have a nonzero mean, while those for the integrated effect method appear to have a near zero mean. These differences, and the approximately four-fold difference in the magnitudes of the standard deviations, is probably due to a flaw in the assumption that the state transition matrix is time invariant for the sample period used.

On the other hand, while the errors are smaller than expected, they can be compared with the results of a similar study (Ref 5) published by Fontana in 1972. He studied strapdown inertial navigation for the European space vehicle. In his study, he examined the effects of sinusoidal vibrations on system accuracy. In his study, he set the g-squared gyro sensitivities at 0.01 degrees/hour/g-squared, while in this study they were generated randomly with a gaussian distribution and a standard deviation of 0.07 degrees/hour/g-squared. His study also differed in that the errors were determined only at the 10 minute point in the flight (orbit injection) and he used only a single sinusoid with results computed for differing vibration amplitudes. He also assumed a worst case in which the angle between the vibration axes and the gyro spin axes was 45 degrees (Ref 5).

Some of the results of Fontana's study are shown in Table 9 on the following page. Vibrations as small as those normally experienced in the C-130A were not examined, but some comparisions can still be made. In his study, the error standard deviations caused by 0.1 g vibrations were exactly two orders of magnitude smaller than those caused by 1.0 g sinusoidal vibrations, indicating that the standard deviations of the induced errors are proportionate to the square of the vibration level. This can be confirmed by examining the g-squared sensitive gyro drift terms in the SIGN-III error model. Equations (44), (45), and (46), which are shown below, are the g-squared sensitive gyro drift terms as extracted from the state equations for states 7, 8, and 9, respectively. Under the assumptions made previously about the nature of the vibration, these are the only terms through which vibration affects system accuracy.

$$x_7(t) = - C_{ex} \, v_n \, v_u \, x_{19}(t) + C_{ey} \, v_e \, v_u \, x_{20}(t)$$

Table 10. RSS Errors Due to Sinusoidal Vibration (Ref 5)

Vibration Amplitude (G's)	Velocity Error (Ft/Sec) X	Velocity Error (Ft/Sec) Y	Velocity Error (Ft/Sec) Z	Position Error (Feet) X	Position Error (Feet) Y	Position Error (Feet) Z
0.1	.0013	.0017	.0013	.3542	.3739	.1345
1.0	.1359	.1708	.1359	35.40	37.36	13.43

Table 11. Comparison of Study Results (Ref 5)

Position Error (Feet)		
Fontana Study (Scaled)	Sinusoidal Method	Integrated Effect Method
.2286	.0996	.0350

$$- C_{ez} v_e v_n x_{21}(t) + OT \qquad (44)$$

$$x_8(t) = - C_{nx} v_n v_u x_{19}(t) + C_{ny} v_e v_u x_{20}(t)$$

$$- C_{nz} v_e v_n x_{21}(t) + OT \qquad (45)$$

$$x_9(t) = - C_{zx} v_n v_u x_{19}(t) + C_{zy} v_e v_u x_{20}(t)$$

$$- C_{zz} v_e v_n x_{21}(t) + OT \qquad (46)$$

where

$x_7(t)$, $x_8(t)$, and $x_9(t)$ = attitude error states, east, north, and up, respectively

$x_{19}(t)$, $x_{20}(t)$, and $x_{21}(t)$ = g-squared gyro drift coefficients

C_{ij} = element of the transformation matrix from local level to SIGN-III coordinates

v_e, v_n, and v_u = vibration in the east, north, and up directions, respectively

OT = other non-g-squared sensitive terms

From these equations, it is evident that vibration in one axis of the local level frame is multiplied by that in another axis in each of the vibration terms. As a result, vibration-induced errors should be proportional to the product of the magnitudes of the vibration. If the vibration levels in each axis are equal, as they were in Reference 5, then the errors would be proportional to the square of the vibration magnitude. The standard deviations of the errors will be affected in similar fashion.

Since the vibration levels and the gyro drift g-squared sensitivities discussed in Reference 5 are not the same as those used in this study, the resulting error standard deviations must be scaled before comparisons are made. This scaling factor was determined as follows:

$$S = (C/Cf) \times (V/Vf) \qquad (47)$$

where

S = the scaling factor

C = the standard deviation of the gyro g-squared sensitive drift coefficients.

Cf = the coefficients used in Fontana's study

Vf = the sum of the squares of the vibration levels used in Fontana's study

$V = \sum A\ B + \sum A\ C + \sum B\ C$

where

A, B, C are the vibration levels used in this study.

The C/Cf term in Equation (47) compensates for the difference in the g-squared gyro drift sensitivities. The V/Vf term is used to correct for the differences in the magnitudes of the sinusoidal vibrations.

Table 10 shows the RSS position errors caused by 0.1 g vibration levels from Fontana's study scaled down by a factor of 2.8 as stated above. The RSS errors from this study are also included. The errors resulting from this study are smaller than those from Fontana's study, but are of the same order of magnitude for the first vibration model and more than one order of magnitude smaller for the second model. Much of the differences in the results can be attributed to Fontana's assumption of worst case orientation of the vibration. Thus, the sinusoidal series method of simulating vibration compares favorably with the results of Fontana's study. However, it is unlikely that Fontana's assumption of worst case conditions can account for the more than order of magnitude difference in the results between his study and those for the experimental integrated effect method. Nonetheless, the computational burden associated with using the sinusoidal series method makes its use impractical unless the mission profile is extremely short. Based on the CPU time required to simulate the 15-minute missions, it woud have required exclusive use of a Cyber 74 computer for over 10 days to complete the simulation of an 8-hour mission. As a result, it could not be used and the full 8-hour mission simulations had to use the integrated effect method in spite of its demonstrated lack of accuracy. The relative efficiency of the this method enabled the full 8-hour mission to be simulated in just under 2 hours. Thus, this method, while not accurate, is at least usable.

Comparison of Errors Due to Vibration and Total System Errors

Figures 26 through 37 show the results of a 20-run, 8-hour simulaton of the entire system model without vibration and similar runs with vibration generated by the experimental

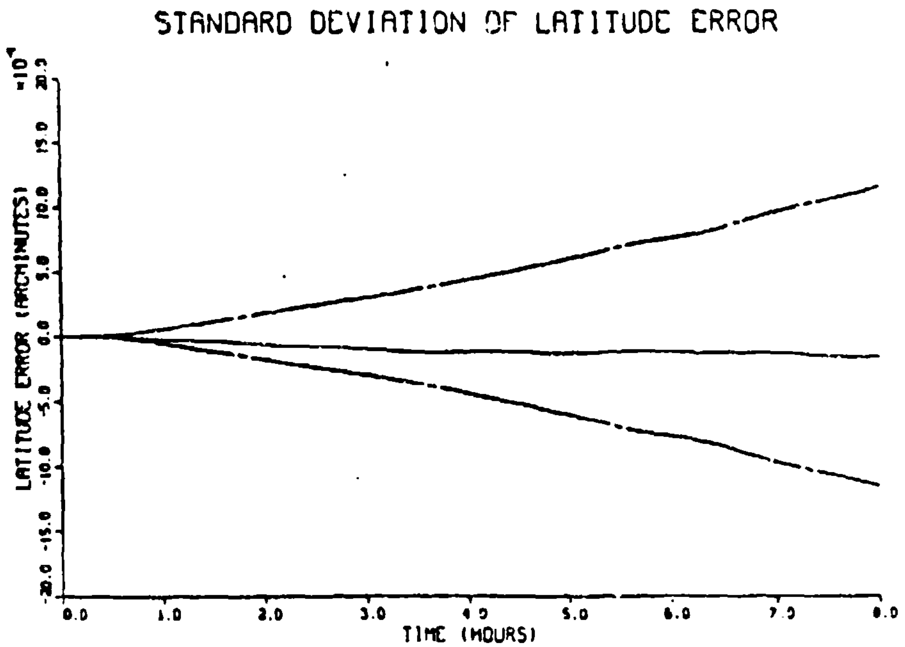


Figure 26. Latitude Error Due to Vibration

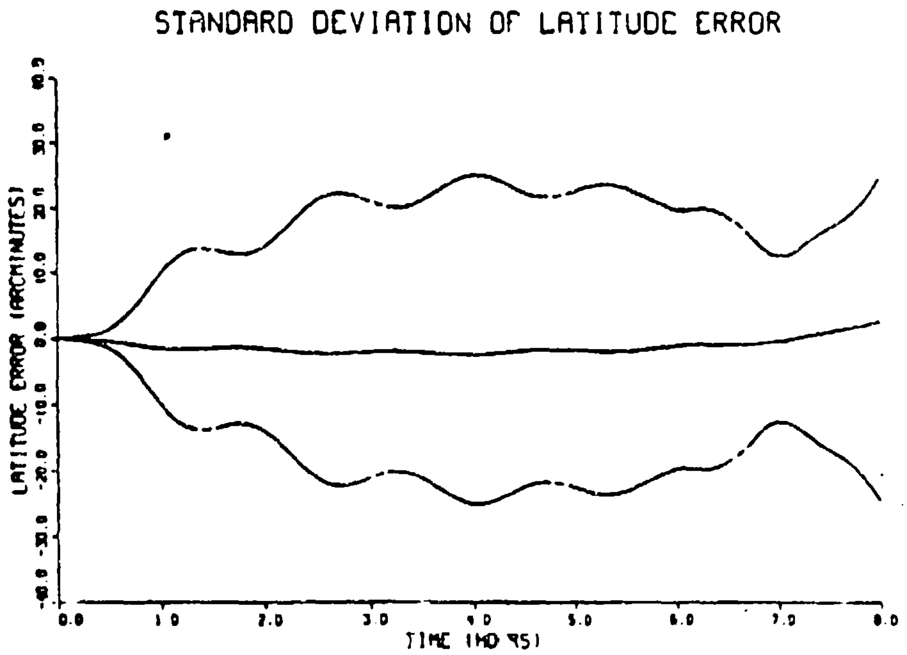


Figure 27. Latitude Error From Non-Vibration Sources

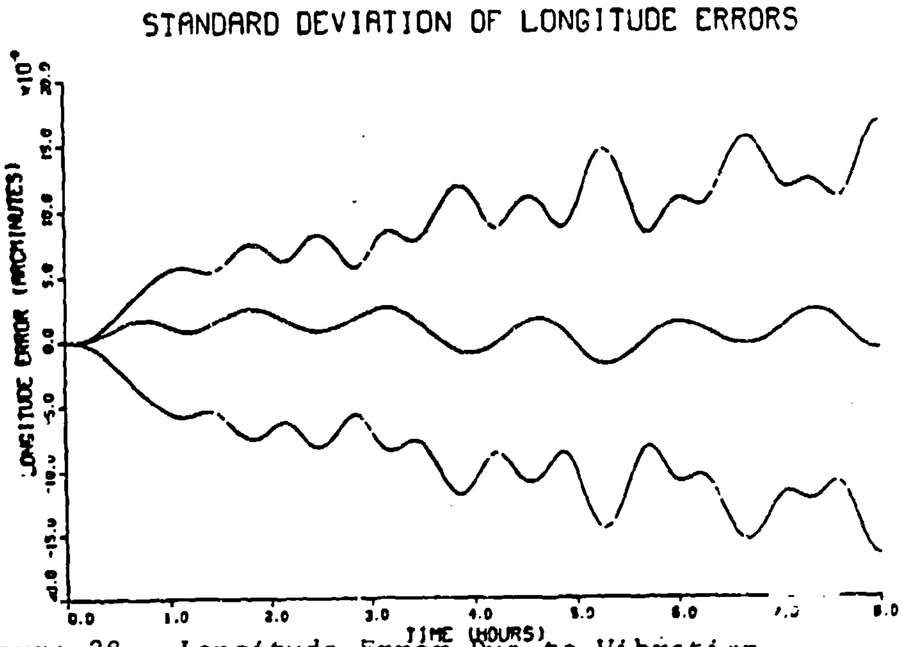


Figure 28. Longitude Error Due to Vibration

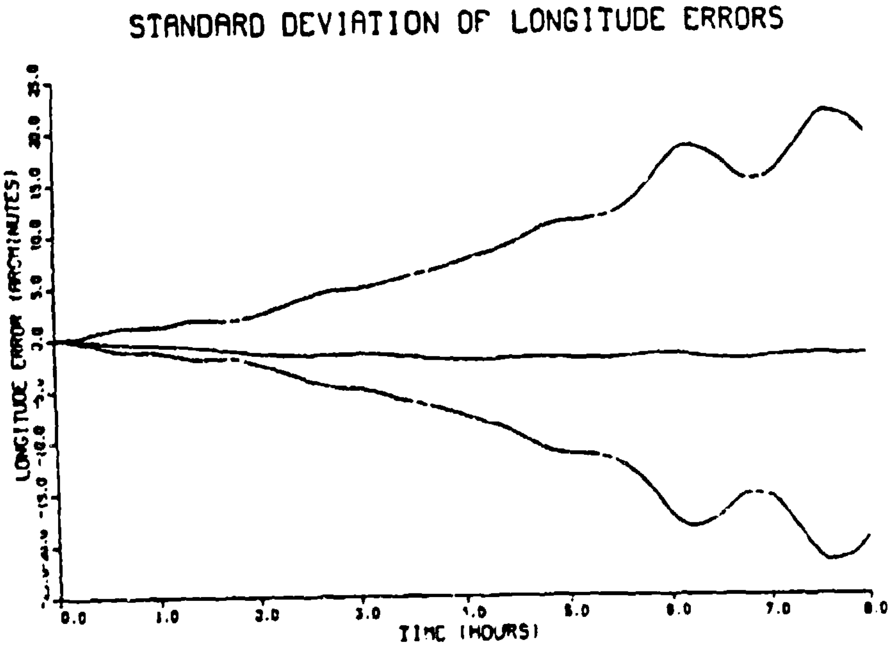


Figure 29. Longitude Error Fron Non-Vibration Sources

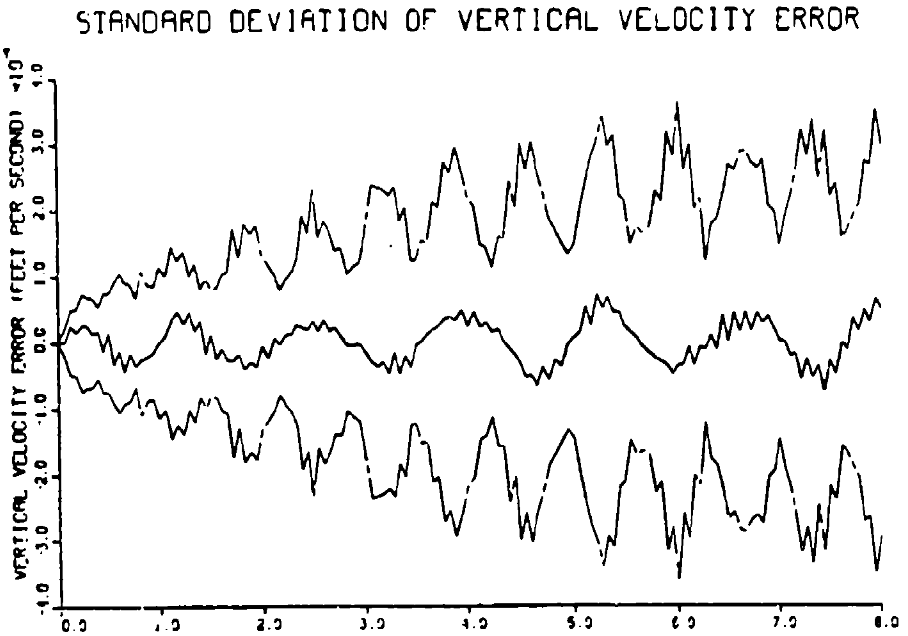


Figure 30. Vert. Vel. Error Due to Vibration

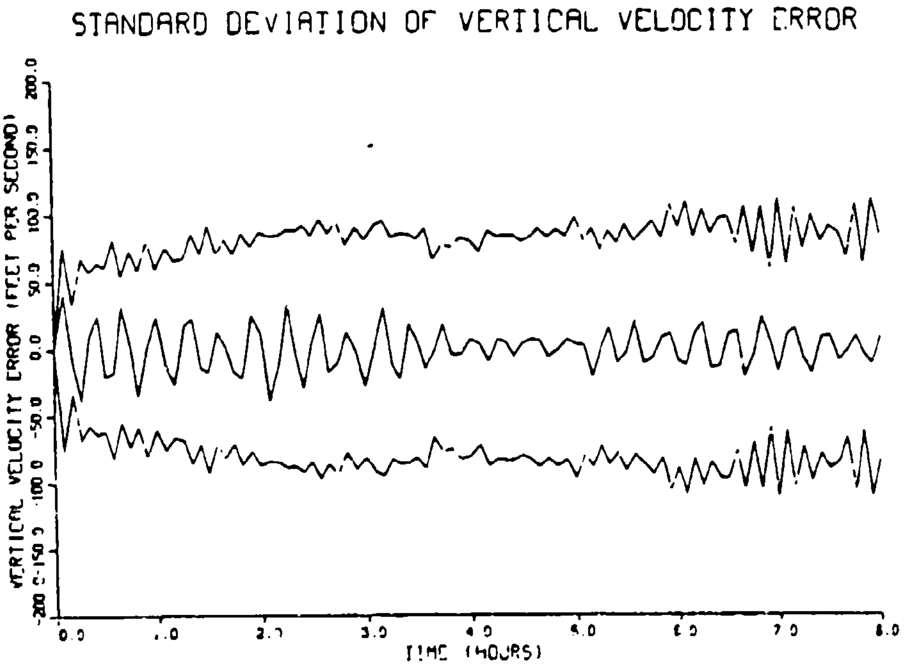


Figure 31. Vert. Vel. Error From Non-Vibration Sources

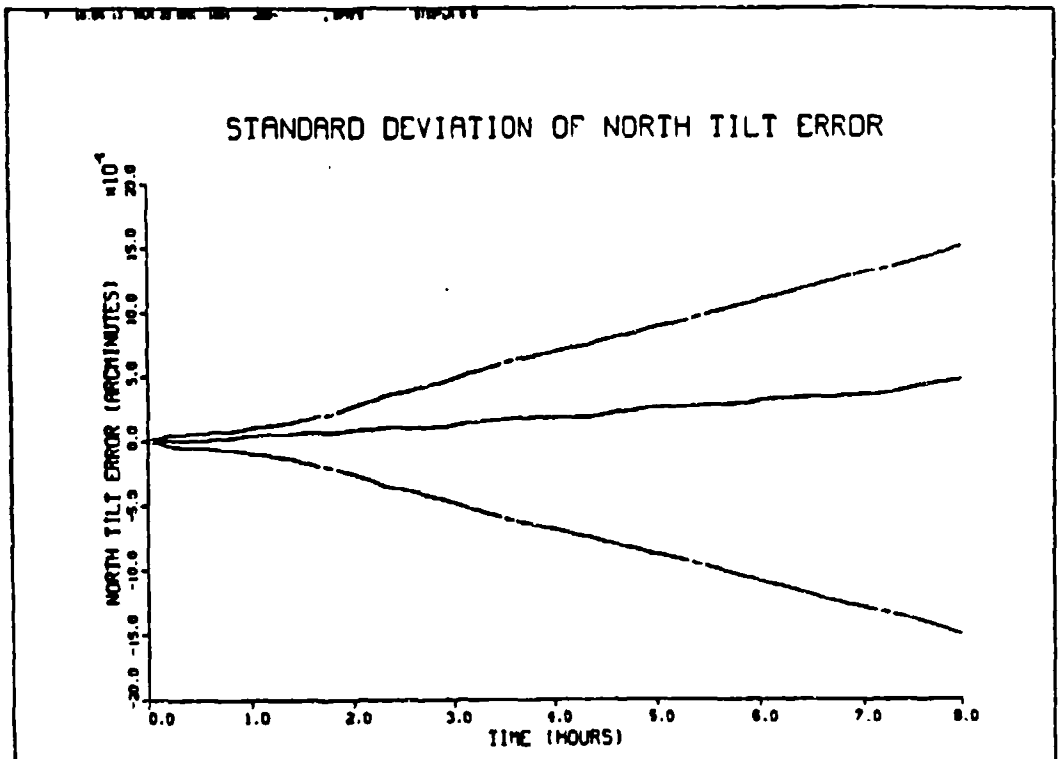


Figure 32. North Tilt Error Due to Vibration

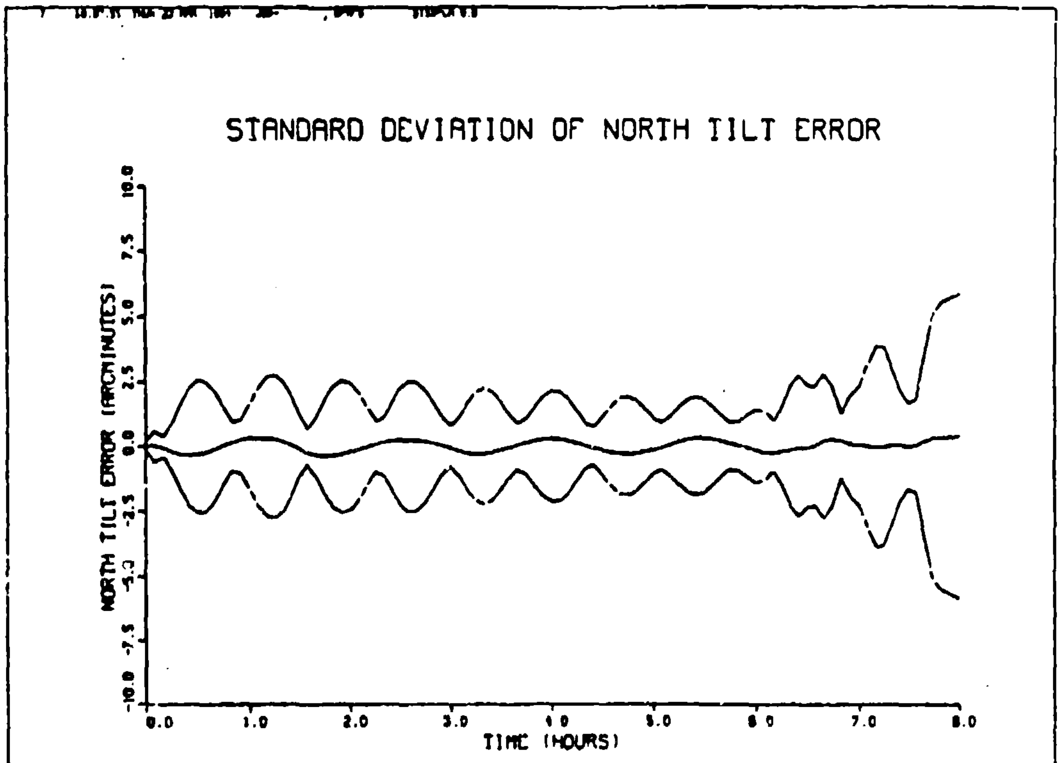


Figure 33. North Tilt Error From Non-Vibration Sources

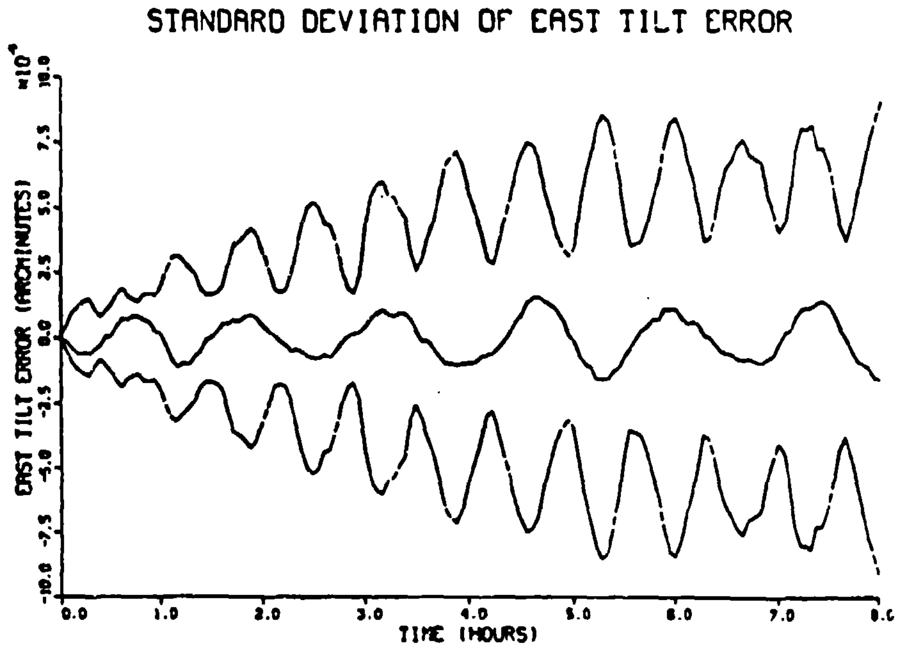


Figure 34. East Tilt Error Due to Vibration

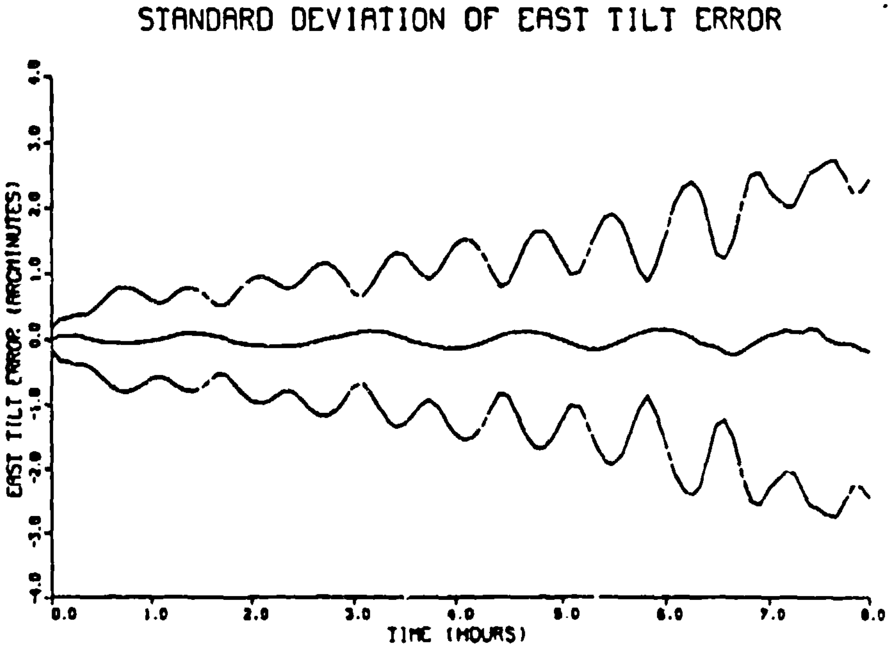


Figure 35. East Tilt Error From Non-Vibration Sources

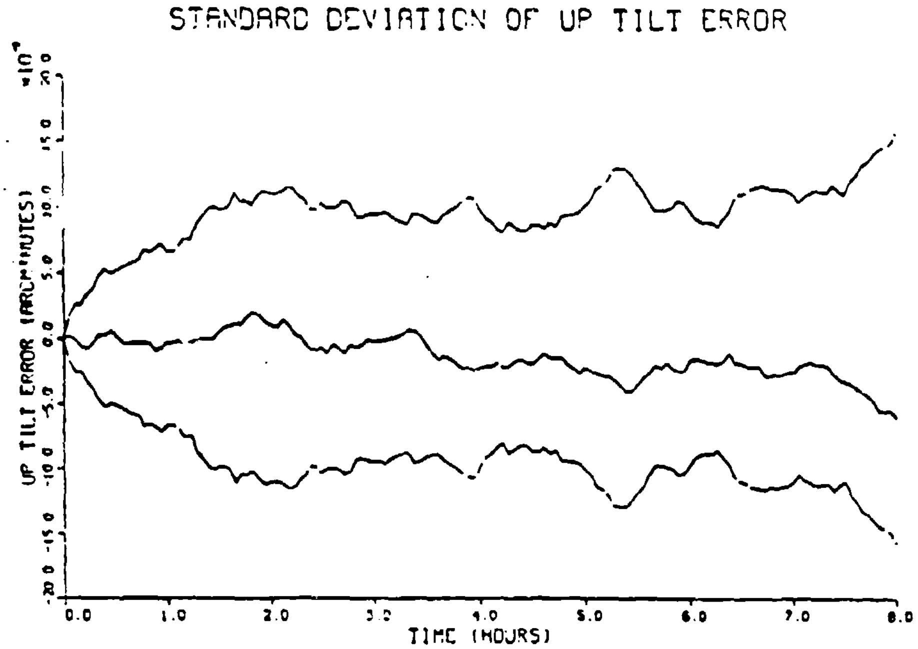


Figure 36. Up Tilt Error Due to Vibration

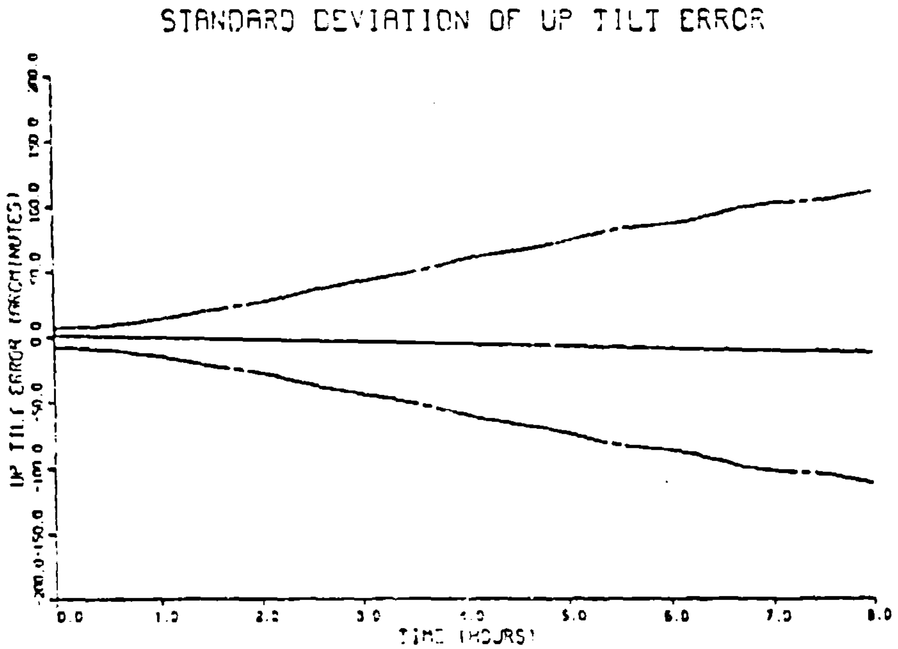


Figure 37. Up Tilt Error From Non-Vibration Sources

integrated effect model with all instrument error sources zeroed out with the exception of g-squared effects. The plots of the vibration runs are at the top of the pages with the total system plots of the same error state at the bottom of the page. While Fontana's study indicates that the second vibration model is not as accurate as the first model, examination of the 15-minute runs of these methods indicates that the second method has some limited utility if the results are scaled up by a factor of approximately five. The resulting loss of accuracy is more than compensated by the resulting 128-fold reduction in cpu time required which makes the second method usable in cases in which the first method is too computationally burdensome. The accuracy of the integrated effect method is too poor to allow its use for anything other than rough order of magnitude estimates of the effects of the vibration. However, it can give an indication of the order of magnitude of the vibration-induced errors and can help in determining if further investigation is warranted.

Since development of both methods was based on the same set of facts and assumptions, with one exception, it is probable that the additional assumption required by the integrated effect model is the cause of the loss of accuracy. This assumption was that the state transition matrix could be treated as time-invariant over the 2 second sampling period. In view of the resulting loss of accuracy, it is likely this assumption was not valid.

However, examination of the results of the full model simulations and of the vibration simulations reveals that, even if the vibration is scaled up by the stated factor of five (note: the plots shown do not include this scaling), the effects of vibration are still several orders of magnitude smaller than the errors caused by other factors. Thus, if either of the vibration models accurately represent the C-130A vibration environment, then airframe vibration is not a major source of inertial navigation system errors.

V. Conclusions and Recommendations

Conclusions

This study examined the effects of airframe vibration on the accuracy of the SIGN-III inertial navigation system. In doing so, it was necessary to make several assumptions. First, it was assumed that the error model given in Widnall and Grundy is accurate and complete except as noted in Chapter 2. It was also assumed that the vibration in each axis was independent in phase from that in the other axes, and that a sinusoidal representation of the vibration was accurate enough to cause minimal error in the study results.

Two vibration models were developed and the results of Monte Carlo simulations of these models are presented. The results of the sinusoidal series method compare favorably with the results of a previous study. Unfortunately, this method is prohibitively burdensome computationally in long simulations involving high frequency vibrations. It required about 1.5 CPU seconds per mission second per Monte Carlo run to simulate this model. The integrated effect method produced errors which were smaller by a factor of about five, but was much more computationally efficient. It ran about 128 times faster than the first method. The results of this method might be usable if the required degree of accuracy is very low and the results are multiplied by a factor of five or if a correcting coefficient is added to correct the error magnitudes. However, at best, the integrated effect method is capable of giving order of magnitude estimates of the vibration-induced errors. As such, it should be used only in determining if further investigation of the vibrational effects is required.

Since development of both methods was based on the same set of facts and assumptions, with one exception, it is probable that the additional assumption required by the integrated effect model is the cause of the loss of accuracy. This assumption was that the state transition matrix could be treated as time-invariant over the 2 second sampling period. In view of the resulting loss of accuracy, it is likely this assumption was not valid.

Regardless of which vibration model is used, the resulting navigation errors are very small, ranging from 4 to 6 orders of magnitude smaller than the total effects of other system error sources.

Recommendations

This study is only a small step in understanding the effects of airframe vibration on inertial navigation system errors and it fell short of its goals. However, the computational benefits of the integrated effect method may merit additional investigation of this technique. In particular, the sampling period should be examined to see if a shorter sampling period will result in improved accuracy. On the other hand, since shortening the sampling period will also increase the computational burden and reduce the utility of the method, it is recommended that sampling periods between 0.1 and 0.5 seconds be used as a starting point. Furthermore, the factor of five correction used with the integrated effect vibration model was obtained by comparing the results of one set of runs produced by the integrated effect model to a similar set produced by the sinusoidal vibration model. No analytical basis for this correction factor was found. Thus, the correction factor used in this study may, or may not, be valid for a different inertial navigation system or a different vibration environment. Future studies should examine the two models to determine an analytical basis for the correction factor.

In addition, studies should be accomplished to determine whether the vibration environment is well represented by a sinusoidal model. It should also be determined to what degree the vibration in each axis is correlated with that in the other axes.

In the system model used in this study, vibration-induced errors entered the system only through the g-squared sensitive gyro drift terms. While it would seem that sinusoidal vibration-induced accelerometer errors should be extremely small, they should be studied to determine if they are significant.

This study was limited to examining only one strapdown inertial navigation system. While the results should be applicable to all systems of comparable type and quality, they may not be applicable to very precise systems used in long range ballistic missiles and strategic bombers such as the B-52 and B-1B. This study was also limited in that no attempt was made to examine the effect of vibration on ring-laser gyro systems. Indepth studies of the effects of vibration on these kinds of systems should be accomplished.

Bibliography

1. Britting, K. R. *Inertial Navigation System Analysis.* New York: John Wiley and Sons, Inc., 1971.

2. Burns, Jay. "Discussions with Mr. Burns, Ref; Vibration Modelling", AFWAL, 8 Nov 1983.

3. Edwards, Robert. "Discussions with Dr Edwards, Ref; Vibration Modelling", AFIT, 3 Nov 1983.

4. Feldmann, Richard E. *SOFEPL: A Plotting Post-Processor for "SOFE", User's Manual.* UDRI-TR-78-88. Dayton Ohio: University of Dayton Research Institute, 1978.

5. Fontana, R. Baldassini. "Strapped Down Inertial Guidance System Study", Paper presented at the Guidance and Control Panel of AGARD meeting, Florence, Italy, October 1972.

6. Grundy, Peter A., and William S. Widnall. *Inertial Navigation System Error Models.* TR-03-73 Technical Report for 6585th Test Group, Holloman AFB, New Mexico. Cambridge, Massachusetts: Intermetrics Incorporated, May 1973.

7. Johnson, Randall G. "Discussions with Capt Johnson, Ref; C-130A Flight Performance", AFIT, 15 September, 1983.

8. Maybeck, Peter S. *"Stochastic Models, Estimation, and Control, Volume I.* New York: Academic Press, 1979.

9. Musick, Stanton H. *PROFGEN-A Computer Program for Generating Flight Profiles.* AFAL-TM-76-247. Wright-Patterson AFB, Ohio: Air Force Avionics Laboratory. November 1976.

10. ______. *SOFE: A Generalized Digital Simulation for Optimal Filter Evaluation User's Manual.* AFAL-TM-78-19. Wright-Patterson AFB, Ohio: Air Force Avionics Laboratory, June 1978.

11. ______. "Discussions with Stanton H. Musick, Ref; Vibration Modelling", AFIT, 9 January, 1984.

12. Parks, Scott E. Standard Pallet Vibration Flight Test. 4950-FTR-82-7. Wright-Patterson AFB, Ohio: 4950 Test Wing, September 1982.

13. Ryan, John E. "Sensitivity Study of Strapdown Inertial Sensors in High Performance Applications." Unpublished MS Thesis. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1980.

Appendix A

SOFE: A Generalized Digital Simulation for Optimal Filter Evaluation (Ref 5)

SOFE is an efficient general purpose program which was developed for the design and evaluation of Kalman filters for use in integrated systems. (Ref 5) Although this thesis did not involve filter design, SOFE was used to perform the Monte Carlo simulation of the Inertial Navigation system error state equations because the software was readily available, provided the necessary numerical precision and efficiency, and included all required simulation capabilities.

SOFE is divided into two modules, basic SOFE and user-written SOFE. Basic SOFE contains 31 routines which perform I/O, problem setup, run setup, numerical integration, measurement update, run termination, and problem termination. User-written SOFE consists of 9 FORTRAN subroutines which specify the problem to be simulated. These subroutines allow for filter state feedback, computation of filter matrices, computation of derivatives, simulation of measurements, trajectory data, etc. SOFE was designed to be efficient in the use of memory and CPU time. This was accomplished by dense packing of arrays and vectors, avoiding the use of double subscripts, and exploiting the symmetry and/or sparse properties of some of the matrices. (Ref 5)

The SOFE truth model is an implementation of the error state equations. Only 4 of the available 9 user-written subroutines were used to implement the truth model. A fifth subroutine, CONVRT, was added for the purpose of converting the units of user inputs to units required for implementation of the truth model. The other subroutines used were USRIN, SNOYS, XSDOT, and TRAJ. USRIN is called only once by SOFE and is used to initialize the problem. In this simulation, it was used to read in numerous constants and the standard deviations of the initial error states. It was also used to call CONVRT. SNOYS was used to simulate the dynamic driving noises by injecting white noise into the solution of the state differential equations of the truth model states at user-specified intervals. XSDOT is used to compute the homogeneous part of the derivative of the truth state vector XS. TRAJ was used only to set some constants and to input trajectory data from PROFGEN.

Additional data are entered through a list called PRDATA in CDC NAMELIST FORMAT. PRDATA includes 40 parameters which remain constant throughout the simulation and are used to specify the user's problem, control I/O, and regulate numerical integration. Listings of all user-written subroutines are included in Appendices C, D, and E.

Appendix B

PROFGEN: A Computer Program for Generating Flight Profiles

PROFGEN (Ref 6) is a computer program which computes flight path data for an aircraft flying a specified route over an ellipsoidal earth. The information provided includes position (geographic latitude, longitude, and altitude), velocity, attitude, and attitude rates of change. Velocity is computed with respect to the earth and is relative to a local vertical (navigation) frame. Acceleration is the sum of the velocity rates of change, gravity, and Coriolis effects. Attitude is expressed in terms of the Euler angles between the path frame and the navigation frame: roll, pitch, and yaw.

PROFGEN models a point mass and does not model the aerodynamics of the aircraft. As a result, the coordinate frame of the flight path is always coincident with the body coordinate frame. The flight profile is composed of up to 50 flight segments, each segment accomplishing a single manuever from the set of five possible manuevers. The maneuvers available are:

Climb or Dive
Coordinated Turns

Sinusoidal Heading Changes
Straight Flights
Rolls

PROFGEN allows for the simulation of various types of aircraft since the user specifies path acceleration rates, and if necessary, centrifugal acceleration rates, for each segment. The final value for each variable in a segment is retained and used as the initial value for the next segment. As a result, an uninterrupted time history exists for each variable.

The earth model used in PROFGEN is a perfect ellipsoid with values for eccentricity, semimajor axis length, spin velocity, and gravitational constant based on the DOD Geodetic System 1972. Modelling of the earth's gravity includes the effects of latitude and altitude changes and has both radial and level components (Ref 6). While this model is not overly precise, it was deemed accurate enough for use, without revision, in this study.

APPENDIX C

SOFE SUBROUTINES FOR 50 STATE MODEL WITHOUT VIBRATION

```
SUBROUTINE SNOYS(IRUN,NF,NS,NXTJ,XF,XS,XTRAJ)
DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ)
COMMON/SNOIS/SDWSO(50),SDWS(10),SDWFO,SDWF
COMMON/DIST/DALT,DGE,DGN,DGU
COMMON/TRJCON/RE,G,OMEGA,RK1,RK2
V=SQRT(XTRAJ(8)**2 + XTRAJ(9)**2)
DT=T-TOLD
SRDT=SQRT(DT)
STDEV=SDWS(1)*SQRT(2*DT*V/DALT)
CORNOS=GAUSS(0.0,STDEV)
XS(6)=XS(6) + RK2*CORNOS
SDEV=(SDQS(2))*SRDT
XS(10)=XS(10) + GAUSS(0.0,SDEV)
SDEV=(SDWS(3))*SRDT
XS(11)=XS(11) + GAUSS(0.0,SDEV)
SDEV=(SDWS(4))*SRDT
XS(12)=XS(12) + GAUSS(0.0,SDEV)
SDEV=(SDWS(5))*SRDT
XS(34)=XS(34) + GAUSS(0.0,SDEV)
SDEV=(SDWS(6))*SRDT
XS(35)=XS(35) + GAUSS(0.0,SDEV)
SDEV=(SDWS(7))*SRDT
XS(36)=XS(36) + GAUSS(0.0,SDEV)
XS(46)=XS(46) + CORNOS
SDEV=(SDWS(8))*SRDT
XS(48)=XS(48) + GAUSS(0.0,SDEV)
SDEV=(SDWS(9))*SRDT
XS(49)=XS(49) + GAUSS(0.0,SDEV)
SDEV=(SDWS(10))*SRDT
XS(50)=XS(50) + GAUSS(0.0,SDEV)
RETURN
ENTRY SNOYSO
TOLD=T
RETURN
END

SUBROUTINE USRIN
COMMON/SNOIS/SDWSO(50),SDWS(10),SDWFO,SDWF
COMMON/DIST/DALT,DGE,DGN,DGU
DIMENSION DIST(4)
NAMELIST/IDIMEN/DIST,NWS,NWSO,SDWSO,SDWS,SDWFO,SDWF
READ(5,IDIMEN)
```

```
WRITE(6,IDIMEN)
DALT=DIST(1)
DGE=DIST(2)
DGN=DIST(3)
DGU=DIST(4)
CALL CONVRT (NWSO,NWS,SDWSO,SDWS)
RETURN
END

SUBROUTINE XFDOT(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ,NTR,PF,XDOT)
DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ),PF(NTR),XDOT(NF)
XDOT(1)=0.0
RETURN
ENTRY XFDOTO
XF(1)=0.0
RETURN
END

SUBROUTINE XSDOT(IRUN,T,NF,NS,NXTJ,XF,XTRAJ,XDOT)
DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ),XDOT(NS)
COMMON/SNOIS/SDWSO(50),SDWS(10),SDWFO,SDWF
COMMON/TRJCON/RE,G,OMEGA,RK1,RK2
COMMON/TRAJ1/V,RLAT,RLON,ALT
COMMON/TRAJ2/VE,VN,VU,FE,FN,FU,WE,WN,WU
COMMON/TRAJ3/CEX,CEY,CEZ,CNX,CNY,CNZ,CUX,CUY,CUZ
COMMON/TRAJ4/SINLAT,COSLAT,TANLAT
COMMON/TRAJ5/OMEGAN,OMEGAU,RHOE,RHON,RHOZ,RKZ
RLAT=XTRAJ(1)
RLON=XTRAJ(2)
ALT=XTRAJ(4)
ROLL=XTRAJ(5)
PITCH=XTRAJ(6)
YAW=XTRAJ(7)
VE=XTRAJ(8)
VN=XTRAJ(9)
VU=XTRAJ(10)
FE=XTRAJ(11)
FN=XTRAJ(12)
FU=XTRAJ(13)
DROLL=XTRAJ(14)
DPITCH=XTRAJ(15)
DYAW=XTRAJ(16)
V=SQRT(VE**2+VN**2)
SINLAT=SIN(RLAT)
COSLAT=COS(RLAT)
TANLAT=SINLAT/COSLAT
SINLON=SIN(RLON)
COSLON=COS(RLON)
OMEGAN=OMEGA*COSLAT
OMEGAU=OMEGA*SINLAT
SX=SIN(YAW)
SY=SIN(ROLL)
```

```
SZ=SIN(PITCH)
CX=COS(YAW)
CY=COS(ROLL)
CZ=COS(PITCH)
RHOE=-VN/RE
RHON=VE/RE
RHOU=VE*TANLAT/RE
WE=RHOE
WN=RHON + OMEGAN
WU=RHOU + OMEGAU
RKZ=VU/RE
F42=2.0*(OMEGAN*VN + OMEGAU*VU) + RHON*VN/COSLAT**2
F43=RHOU*RHOE + RHON*RKZ
F44=-RHOE*TANLAT - RKZ
F52=-2.0*OMEGAN8VE - RHON*VE/COSLAT**2
F53=RHON*RHOU - RHOE*RKZ
F63=2.0*G/RE - RHON**2 - RHOE**2
F92=WN + RHOU*TANLAT
CXE=SX
CXN=SZ*CY
CXU=CZ*CY
CYE=-CY*SX
CYN=SZ*SY*SX + CZ*CX
CYU=CZ*SY*SX - SZ*CX
CZE=-CY*CX
CZN=-CZ*SX + SZ*SY*CX
CZU=CZ*SY*CX + SY*SX
CEX=CXE
CEY=CYE
CEZ=CZE
CNX=CXN
CNY=CYN
CZY=CZN
CUX=CXU
CUY=CYU
CUZ=CZU
FX=CXN*FN + CXE*FE + CXU*FU
FY=CYN*FN + CYE*FE + CYU*FU
FZ=CZN*FN + CZE*FE + CZU*FU
WX=CXN*WN + CXE*WE + CXU*WU + DYAW
WY=CYN*WN + CYE*WE + CYU*WU + DROLL
WZ=CZN*WN + CZE*WE + CZU*WU + DPITCH
WXP=0.0
WXM=0.0
WYP=0.0
WYM=0.0
WZP=0.0
WZM=0.0
IF (WX .GE. 0.0 ) WXP=WX
IF (WX .LE. 0.0 ) WXM=WX
IF (WY .GE. 0.0 ) WYP=WX
IF (WY .LE. 0.0 ) WYM=WX
```

```
      IF (WZ .GE. 0.0 ) WZP=WZ
      IF (WZ .LE. 0.0 ) WZM=WZ
      XDOT(1)=XS(2)*RHOU/COSLAT - XS(3)*RHON/(RE*COSLAT)
     1         + XS(4)/(RE*COSLAT)
      XDOT(2)=XS(3)*RHOE/RE + XS(5)/RE
      XDOT(3)=XS(6)
      XDOT(4)=XS(2)*F42 + XS(3)*F43 + XS(4)*F44
     1         + XS(5)*(WU + OMEGAU) - XS(6)*(WN + OMEGAN)
     1         - XS(8)*FU + XS(9)*FN + XS(34)*CEX
     1    + XS(35)*CEY + XS(36)*CEZ
     1         + XS(37)*CEX*FX + XS(38)*CEY*FY
     1         + XS(39)*CEZ*FZ - XS(40)*CEX*FZ
     1         + XS(41)*CEX*FY + XS(42)*CEY*FZ
     1         - XS(43)*CEY*FX - XS(44)*CEZ*FY
     1         + XS(45)*CEZ*FX + XS(48)
      XDOT(5)=XS(2)*F52 + XS(3)*F53 - XS(4)*2.0*WU - XS(5)*RKZ
     1         + XS(6)*RHOE + XS(7)*FU - XS(9)*FE
     1         + XS(34)*CNX + XS(35)*CNY + XS(36)*CNZ
     1         + XS(37)*CNX*FX + XS(38)*CNY*FY
     1         + XS(39)*CNZ*FZ - XS(40)*CNX*FZ
     1         + XS(41)*CNX*FY + XS(42)*CNY*FZ - XS(43)*CNY*FX
     1         - XS(44)*CNZ*FY + XS(45)*CNZ*FX + XS(49)
      XDOT(6)= - XS(2)*2.0*OMEGAU*VE + XS(3)*(F63 - RK1)
     1         + XS(4)*2.0*WN - XS(5)*2.0*RHOE - XS(6)*RK2
     1         - XS(7)*FN + XS(8)*FE + XS(34)*CUX
     1         + XS(35)*CUY + XS(36)*CUZ + XS(37)*CUZ*FX
     1         + XS(38)*CUY*FY + XS(39)*CUZ*FZ
     1         - XS(40)*CUX*FZ + XS(41)*CUX*FY
     1         + XS(42)*CUY*FZ - XS(43)*CUY*FX
     1         - XS(44)CUZ*FY + XS(45)*CUZ*FX
     1         + XS(46)*(RK1 - RK2 + V/DALT)
     1         + XS(47)*(RK1*ALT + RK2*VU)
     1         + XS(50)
      XDOT(7)= - XS(3)*RHOE/RE - XS(5)/RE + XS(8)*WU
     1         - XS(9)*WN + XS(10)*CEX + XS(11)*CEY
     1         + XS(12)*CEZ + XS(13)*CEX*FX
     1         + XS(14)*CEX*FZ + XS(15)*CEY*FY
     1         + XS(16)*CEY*FZ + XS(17)*CEZ*FZ
     1         - XS(18)*CEZ*FY - XS(19)*CEX*FX*FY
     1         + XS(20)*CEY*FX*FZ - XS(21)*CEZ*FX*FY
     1         + XS(22)*CEX*WXP + XS(23)*CEX*WXM
     1         + XS(24)*CEY*WYP + XS(25)*CEY*WYM
     1         + XS(26)*CEZ*WZP + XS(27)*CEZ*WZM
     1         + XS(28)*CEX*WZ - XS(29)*CEX*WY
     1         - XS(30)*CEY*WZ + XS(31)*CEY*WX
     1         + XS(32)*CEZ*WY - XS(33)*CEZ*WX
      XDOT(8)= - XS(2)*OMEGAU -XS(3)*RHON/RE + XS(4)/RE
     1         - XS(7)*WU + XS(9)*WE + XS(10)*CNX
     1         + XS(11)*CNY + XS(12)*CNZ + XS(13)*CNX*FX
     1         + XS(14)*CNX*FZ + XS(15)CNY*FY
     1         + XS(16)*CNY*FZ + XS(17)*CNZ*FZ
     1         - XS(18)*CNZ*FY - XS(19)*CNX*FY*FZ
     1        + XS(20)*CNY*FX*FZ - XS(21)*CNZ*FX*FY
     1        + XS(22)*CNX*WXP + XS(23)*CNX*WXM
     1        + XS(26)*CNZ*WZP + XS(27)*CNZ*WZM
     1        + XS(28)*CNX*WZ - XS(29)*CNX*WY
     1        - XS(30)*CNY*WZ + XS(31)*CNY*WX
     1        + XS(32)*CNZ*WY - XS(33)*CNZ*WX
      XDOT(9)= XS(2)*F92 - XS(3)*RHOU/RE + XS(4)*TANLAT/RE
     1        + XS(7)*WN - XS(8)*WE + XS(10)*CUX + XS(11)*CUY
     1        + XS(12)*CUZ + XS(13)*CUX*FX + XS(14)*CUX*FZ
     1        + XS(15)*CUY*FY + XS(16)*CUY*FZ
     1        + XS(17)*CUZ*FZ - XS(18)*CUZ*FY
     1        - XS(19)*CUX*FY*FZ + XS(20)*CUY*FX*FZ
     1        - XS(21)*CUZ*FX*FY + XS(22)*CUX*WXP
     1        + XS(23)*CUX*WXM + XS(24)*CUY*WYP
     1        + XS(25)*CUY*WYM + XS(26)*CUZ*WZP
     1        + XS(27)*CUZ*WZM + XS(28)*CUX*WZ
     1        - XS(29)*CUX*WY - XS(30)*CUY*WZ
     1        + XS(31)*CUY*WX + XS(32)*CUZ*WY
     1        - XS(33)*CUZ*WX
      DO 10 I=10,45
   10 XDOT(I)=0.0
      XDOT(46)= - XS(46)*V/DALT
      XDOT(47)=0.0
      XDOT(48)= - XS(48)*V/DGE
      XDOT(49)= - XS(49)*V/DGN
      XDOT(50)= - XS(50)*V/DGU
      RETURN
      ENTRY XSDOTO
      ALPHA=XTRAJ(3)
      DO 25 I=1,6
   25 XS(I)=GAUSS(0.0,SDWSO(I))
      DO 45 I=10,50
   45 XS(I)=GAUSS(0.0,SDWSO(I))
      RLAT=XTRAJ(1)
      OMEGA=7.2921151E-5
      OMEGAN=OMEGA*COS(RLAT)
      OMEGAU=OMEGA*SIN(RLAT)
      RK1=3E-2
      RK2=3E-4
      G=32.0881576
      RE=20925640.0
      RANHED=GAUSS(0.0,1.8138)
      CHEAD=COS(RANHEAD)
      SHEAD=SIN(RHEAD)
      CNX=-CHEAD
      CUX=SHEAD
      SUY=CHEAD
      CNY=SHEAD
      WX=CNX*OMEGAN + CUX*OMEGAU
      WY=CUY*OMEGAU + CNY*OMEGAU
      XS(7)=XS(4)*2.*OMEGAU/G - XS(6)*2.*OMEGAN/G
     1        + XS(36)/G - XS(45) + XS(48)/G
     XS(8)=XS(5)*2.*OMEGAU/G - XS(6)*2.*OMEGAN/G
    1        + XS(36)/G - XS(45) + XS(48)/G
     XS(9)=( - XS(5)/RE + XS(8)*OMEGAU + XS(12)
    1        + XS(31)*CNY*WX)/OMEGAU
     RETURN
     END

SUBROUTINE TRAJO(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ)
     DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ)
     COMMON/TRJCON/RE,G,OMEGA,RK1,RK2
     INTEGER ITITLE(60),IPRSET(19),IRTSET(19)
     NAMELIST/PRDATA/NSEGT,TSTART,VTO,ROLLO,PITCHO,HEADO,ALFAO
    1     GLATO,TLONO,ALTO,IPRNT,IRITE,IPLOT,ROLRAT,ROLTC,
    1     LLMECH,LUNIT,RELERR,ABSERR,IPRSET,IRTSET
C   READ AND ECHO TITLE AND INPUT DATA FROM PROFGEN (TAPE 3)
     READ(3) ITITLE,TODAY,CLOCK
     READ(3) NSEGT,TSTART,VTO,ROLLO,PITCHO,HEADO,ALFAO
    1        GLATO,TLONO,ALTO,IPRNT,IRITE,IPLOT,ROLRAT,
    1        ROLTC,LLMECH,LUNIT,RELERR,ABSERR,IPRSET,IRTSET
     DO 10 I=1,16
 10  READ(3) DUM
     IF (IRUN .NE. 1) RETURN
     WRITE(6,PRDATA)
     WRITE(6,100) (ITITLE(I),I=1,6),TODAY,CLOCK
     OMEGA=70292115E-5
     G=32.12698510
     RE=20925640.0
     RK1=3.0E-2
     RK2=3.0E-4
 100 FORMAT (//5X,"THE ABOVE DATA WAS USED TO CREATE THE TRAJECTORY
    *USING 'PROFGEN':"
    *      //10X,"TRAJECTORY TITLE:  ",6A10,
    *      /10X,TRAJECTORY RUN DATE AND TIME: "A10,5X,A10)
     RETURN
     END

     SUBROUTINE CONVRT(NWSO,NWS,SDWSO,SDWS)
     DIMENSION SDWSO(NWSO),SDWS(NWS)
     PI=ABS(ACOS(-1.0))
 C    DEGREES TO RADIANS CONVERSION
     CONV1=PI/180.0
 C    HOURS TO SECONDS CONVERSION
     CONV2=3600.0
 C    ARCMINUTES TO RADIANS CONVERTION
     CONV3=(1.0/60.0)*CONV1
 C    ARCSECONDS TO RADIANS CONVERSION
     CONV4=(1.0/60.0)*CONV3
 C    G'S TO FEET PER SECOND SQUARED CONVERSION
     CONV5=32.2
 C    MICRO G'S TO FEET PER SECOND SQUARED CONVERSION
      CONV6=32.2E-6
 C     DEGREES PER HOUR TO RADIANS PER SECOND CONVERSION
```

```
      CONV7=CONV1/CONV2
C      PARTS PER MILLION CONVERSION
      CONV8=1.0E-6
C      MICRORADIANS TO RADIANS CONVERSION
      CONV9=1.0E-6
C      G SQUARED TO (FEET PER SECOND SQUARED) CONVERSION
      CONV10=32.2**2
C      SQUARE ROOT HOUR TO SQUARE ROOT SECOND CONVERSION
      CONV11=60.0
C      NAUTICAL MILES TO FEET CONVERSION
      CONV12=6076.10333
      DO 15 I=1,2
   15 SDWSO(I)=SDWSO(I)*CONV3
      DO 20 I=10,12
   20 SDWSO(I)=SDWSO(I)*CONV7
      DO 40 I=13,18
   40 SDWSO(I)=SDWSO(I)*CONV7/CONV5
      DO 50 I=19,21
   50 SDWSO(I)=SDWSO(I)*CONV7/CONV5**2
      DO 60 I=22,27
   60 SDCWSO(I)=SDWSO(I)*CONV8
      DO 70 I=28,33
   70 SDWSO(I)=SDWSO(I)*CONV4
      DO 80 I=34,36
   80 SDWSO(I)=SDWSO(I)*CONV6
      DO 90 I=37,39
   90 SDWSO(I)=SDWSO(I)*CONV8
      DO 100 I=40,45
  100 SDWSO(I)=SDWSO(I0*CONV6
      DO 110 I=48,50
  110 SDWSO(I)=SDWSO(I)*CONV6
      DO 120 I=2,4
  120 SDWS(I)=SDWS(I)*CONV1/(CONV2*CONV11)
      DO 130 I=5,7
  130 SDWS(I)=SDWS(I)*CONV6/CONV11
      DO 140 I=8,10
  140 SDWS(I)=SDWS(I)*CONV6
      RETURN
      END
```

APPENDIX D

SOFE SUBROUTINES FOR SINUSOIDAL VIBRATION RUN

```
      SUBROUTINE SNOYS(IRUN,NF,NS,NXTJ,XF,XS,XTRAJ)
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ)
      DIMENSION PSI(9),PHI(9),THETA(9),AX(9),AY(9),AZ(9)
      DIMENSION AXY(9),AXZ(9),AYZ(9)
      COMMON/SNOIS/FU,CEX,CEY,CEZ,CNX,CNY,CNZ,CUX,CUY,CUZ
      COMMON/SNOIS2/FXX,FYY,FZZ
      DT=T-TOLD
      WT=320*T
      FX=FXX
      FY=FYY
      FZ=FZZ
      DO 10 I=1,9
      FX=FX + COS(WT*I + PSI(I))*AX(I)
      FY=FY + COS(WT*I + PHI(I))*AY(I)
      FZ=FZ + COS(WT*I + THETA(I))*AZ(I)
      FXFY=FX*FY
      FXFZ=FX*FZ
      FYFZ=FY*FZ
      V1= - DX*CEX*FYFZ + DY*CEY*FXFZ - DZ*CEZ*FXFY
      V2= - DX*CNX*FYFZ + DY*CNY*FXFZ - DZ*CNZ*FXFY
      V3= - DX*CUX*FYFZ + DY*CUY*FXFZ - DZ*CUZ*FXFY
      XS(7)=XS(7) + V1*DT
      XS(8)=XS(8) + V2*DT
      XS(9)=XS(9) + V3*DT
      RETURN
      ENTRY SNOYSO
      DO 30 I=1,9
      PHI(I)=GAUSS(0.0,1.81)
      PSI(I)=GAUSS(0.0,1.81)
      THETA(I)=GAUSS(0.0,1.81)
   30 CONTINUE
      AX(1)=0.2478
      AX(2)=0.2655
      AX(3)=0.223
      AX(4)=0.112
      AX(5)=0.204
      AX(6)=0.112
      AX(7)=0.129
      AX(8)=0.170
      AX(9)=0.091
      AY(1)=0.5657
      AY(2)=0.2325
      AY(3)=0.273
      AY(4)=0.170
      AY(5)=0.011
      AY(6)=0.112
      AY(7)=0.129
      AY(8)=0.158
      AY(9)=0.091
      AZ(1)=1.8238
      AZ(2)=0.864
      AZ(3)=0.815
      AZ(4)=0.288
      AZ(5)=0.407
      AZ(6)=0.500
      AZ(7)=0.644
      AZ(8)=0.288
      AZ(9)=0.204
      DO 40 I=1,9
      DX=GAUSS(0.0,3.273E-10)
      DY=GAUSS(0.0,3.273E-10)
      DZ=GAUSS(0.0,3.273E-10)
      TOLD=T
      RETURN
      END

      SUBROUTINE USRIN
      RETURN
      END

      SUBROUTINE XFDOT(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ,NTR,PF,XDOT)
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ),PF(NTR),XDOT(NF)
      XDOT(1)=0.0
      RETURN
      ENTRY XFDOTO
      XF(1)=0.0
      RETURN
      END

      SUBROUTINE XSDOT(IRUN,T,NF,NS,NXTJ,XF,XTRAJ,XDOT)
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ),XDOT(NS)
      COMMON/TRJCON/RE,G,OMEGA,RK1,RK2
      COMMON/SNOIS2/FXX,XYY,FZZ
      COMMON/SNOIS/FU,CEX,CEY,CEZ,CNY,CNY,CNZ,CUX,CUY,CUZ
      XDOT(1)= XS(4)/(RE*COSLAT)
      XDOT(2)= XS(5)/RE
      XDOT(3)= XS(6)
      XDOT(4)= XS(5)* 2.0 * OMEGAU - XS(6)* 2.0 * OMEGAN
     1        - XS(8)* G
      XDOT(5)= - XS(4)*2.0*WU
      XDOT(6)= XS(3)*(2.0*G/RE-RK1) + XS(4)*2.0*OMEGAN
     1        - XS(6)*RK2
      XDOT(7)= - XS(5)/RE + XS(8)*OMEGAU - XS(9)*OMEGAN
      XDOT(8)= - XS(2)*OMEGAU* + XS(4)/RE - XS(7)*OMEGAU
      XDOT(9)= XS(2)*OMEGAN + XS(4)*TANLAT/RE + XS(7)*OMEGAN
      RETURN
      ENTRY XSDOTO
      ROLL=XTRAJ(5)
      PITCH=XTRAJ(6)
      YAW=XTRAJ(7)
      FU=XTRAJ(13)
      SX=SIN(YAW)
      SY=SIN(ROLL)
      SZ=SIN(PITCH)
      CX=COS(YAW)
      CY=COS(ROLL)
      CZ=COS(PITCH)
      CEX=SX
      CEY= -CY*SX
      CEZ= -CY*CX
      CNX=SZ*CY
      CNY=SZ*SY*SX + CZ*CX
      CNZ= -CZ*SX +SZ*SY*CX
      CUX=CZ*CY
      CUY=CZ*SY*CX + SZ*CX
      CUZ=CZ*SY*CX + SZ*SX
      FXX=CUX*FU
      FYY=CUY*FU
      FZZ=CUX*FU
      ALPHA=XTRAJ(3)
      DO 25 I=1,9
 25   XS(I)=0.0
      RLAT=XTRAJ(1)
      COSLAT=COS(RLAT)
      TANLAT=TAN(RLAT)
      OMEGA=7.2921151E-5
      OMEGAN=OMEGA*COSLAT
      OMEGAU=OMEGA*SINLAT
      RK1=3.0E-2
      RK2=3.0E-4
      G=32.0881576
      RE=20925640.0
      SDEV=1.0
      DX=GAUSS(0.0,3.273E-10)
      DY=GAUSS(0.0,3.273E-10)
      DZ=GAUSS(0.0,3.273E-10)
      RETURN
      END

SUBROUTINE TRAJO(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ)
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ)
      COMMON/TRJCON/RE,G,OMEGA,RK1,RK2
      INTEGER ITITLE(60),IPRSET(19),IRTSET(19)
      NAMELIST/PRDATA/NSEGT,TSTART,VTO,ROLLO,PITCHO,HEADO,ALFAO
     1     GLATO,TLONO,ALTO,IPRNT,IRITE,IPLOT,ROLRAT,ROLTC,
     1     LLMECH,LUNIT,RELERR,ABSERR,IPRSET,IRTSET
C   READ AND ECHO TITLE AND INPUT DATA FROM PROFGEN (TAPE 3)
```

```
      READ(3) ITITLE,TODAY,CLOCK
      READ(3) NSEGT,TSTART,VTO,ROLLO,PITCHO,HEADO,ALFAO
     1        GLATO,TLONO,ALTO,IPRNT,IRITE,IPLOT,ROLRAT,
     1        ROLTC,LLMECH,LUNIT,RELERR,ABSERR,IPRSET,IRTSET
      DO 10 I=1,16
 10   READ(3) DUM
      IF (IRUN .NE. 1) RETURN
      WRITE(6,PRDATA)
      WRITE(6,100)(ITITLE(I),I=1,6),TODAY,CLOCK
      OMEGA=70292115E-5
      G=32.12698510
      RE=20925640.0
      RK1=3.0E-2
      RK2=3.0E-4
 100  FORMAT (//5X,"THE ABOVE DATA WAS USED TO CREATE THE TRAJECTORY
     *USING 'PROFGEN':"
     *       //10X,"TRAJECTORY TITLE:  ",6A10,
     *       /10X,TRAJECTORY RUN DATE AND TIME: "A10,5X,A10)
      RETURN
      END
```

APPENDIX E

SOFE SUBROUTINES FOR 50 STATE MODEL WITH VIBRATION

```
      SUBROUTINE SNOYS(IRUN,NF,NS,NXTJ,XF,XS,XTRAJ)
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ)
      DIMENSION PSI(9),PHI(9),THETA(9),AX(9),AY(9),AZ(9)
      DIMENSION AXY(9),AXZ(9),AYZ(9)
      COMMON/SNOIS/SDWSO(50),SDWS(10),SDWFO,SDWF
      COMMON/TRAJ2/VE,VN,VU,FE,FN,FU,WE,WN,WU
      COMMON/TRAJ3/CEX,CEY,CEZ,CNX,CNY,CNZ,CUX,CUY,CUZ
      COMMON/DIST/DALT,DGE,DGN,DGU
      COMMON/TRJCON/RE,G,OMEGA,RK1,RK2
      V=SQRT(XTRAJ(8)**2 + XTRAJ(9)**2)
      DT=T-TOLD
      SRDT=SQRT(DT)
      STDEV=SDWS(1)*SQRT(2*DT*V/DALT)
      CORNOS=GAUSS(0.0,STDEV)
      XS(6)=XS(6) + RK2*CORNOS
      SDEV=(SDQS(2))*SRDT
      XS(10)=XS(10) + GAUSS(0.0,SDEV)
      SDEV=(SDWS(3))*SRDT
      XS(11)=XS(11) + GAUSS(0.0,SDEV)
      SDEV=(SDWS(4))*SRDT
      XS(12)=XS(12) + GAUSS(0.0,SDEV)
      SDEV=(SDWS(5))*SRDT
      XS(34)=XS(34) + GAUSS(0.0,SDEV)
      SDEV=(SDWS(6))*SRDT
      XS(35)=XS(35) + GAUSS(0.0,SDEV)
      SDEV=(SDWS(7))*SRDT
      XS(36)=XS(36) + GAUSS(0.0,SDEV)
      XS(46)=XS(46) + CORNOS
      SDEV=(SDWS(8))*SRDT
      XS(48)=XS(48) + GAUSS(0.0,SDEV)
      SDEV=(SDWS(9))*SRDT
      XS(49)=XS(49) + GAUSS(0.0,SDEV)
      SDEV=(SDWS(10))*SRDT
      XS(50)=XS(50) + GAUSS(0.0,SDEV)
      DO 10 I=1,9
      PSI(I)=PSI(I) + GAUSS(0.0,PHASE)
      PHI(I)=PHI(I) + GAUSS(0.0,PHASE)
      THETA(I)=THETA(I) + GAUSS(0.0,PHASE)
10    CONTINUE
      DX=XS(19)
      DY=XS(20)
      DZ=XS(21)
      DXCEX=DX*CEX
      DYCEY=DY*CEY
      DZCEZ=DZ*CEZ
      DXCNX=DX*CNX
      DYCNY=DY*CNY
      DZCNZ=DZ*CNZ
      DXCUX=DX*CUX
      DYCUY=DY*CUY
      DZCUZ=DZ*CUZ
      V1=0.0
      V2=0.0
      V3=0.0
      DO 20 I=1,9
      V1=V1 + ( - DXCEX * AYZ(I) * COS(PHI(I)-THETA(I))
     1          + DYCEY * AXZ(I) * COS(PSI(I) - THETA(I))
     1          - DZCEZ * AXY(I) * COS(PSI(I)-PHI(I)))/2
      V2=V2 + ( - DXCNX * AYZ(I) * COS(PHI(I)-THETA(I))
     1          + DYCNY * AXZ(I) * COS(PSI(I) - THETA(I))
     1          - DZCNZ * AXY(I) * COS(PSI(I)-PHI(I)))/2
      V3=V3 + ( - DXCUX * AYZ(I) * COS(PHI(I)-THETA(I))
     1          + DYCUY * AXZ(I) * COS(PSI(I) - THETA(I))
     1          - DZCUZ * AXY(I) * COS(PSI(I)-PHI(I)))/2
20    CONTINUE
      XS(7)=XS(7) + V1*DT
      XS(8)=XS(8) + V2*DT
      XS(9)=XS(9) + V3*DT
      RETURN
      ENTRY SNOYSO
      DO 30 I=1,9
      PHI(I)=GAUSS(0.0,1.81)
      PSI(I)=GAUSS(0.0,1.81)
      THETA(I)=GAUSS(0.0,1.81)
30    CONTINUE
      AX(1)=0.2478
      AX(2)=0.2655
      AX(3)=0.223
      AX(4)=0.112
      AX(5)=0.204
      AX(6)=0.112
      AX(7)=0.129
      AX(8)=0.170
      AX(9)=0.091
      AY(1)=0.5657
      AY(2)=0.2325
      AY(3)=0.273
      AY(4)=0.170
      AY(5)=0.011
      AY(6)=0.112
      AY(7)=0.129
      AY(8)=0.158
      AY(9)=0.091
      AZ(1)=1.8238
      AZ(2)=0.864
      AZ(3)=0.815
      AZ(4)=0.288
      AZ(5)=0.407
      AZ(6)=0.500
      AZ(7)=0.644
      AZ(8)=0.288
      AZ(9)=0.204
      DO 40 I=1,9
      AXY(I)=AX(I)*AY(I)
      AXZ(I)=AX(I)*AZ(I)
      AYZ(I)=AY(I)*AZ(I)
  40  CONTINUE
      TOLD=T
      RETURN
      END

      SUBROUTINE USRIN
      COMMON/SNOIS/SDWSO(50),SDWS(10),SDWFO,SDWF
      COMMON/DIST/DALT,DGE,DGN,DGU
      DIMENSION DIST(4)
      NAMELIST/IDIMEN/DIST,NWS,NWSO,SDWSO,SDWS,SDWFO,SDWF
      READ(5,IDIMEN)
      WRITE(6,IDIMEN)
      DALT=DIST(1)
      DGE=DIST(2)
      DGN=DIST(3)
      DGU=DIST(4)
      CALL CONVRT (NWSO,NWS,SDWSO,SDWS)
      RETURN
      END

      SUBROUTINE XFDOT(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ,NTR,PF,XDOT)
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ),PF(NTR),XDOT(NF)
      XDOT(1)=0.0
      RETURN
      ENTRY XFDOTO
      XF(1)=0.0
      RETURN
      END

      SUBROUTINE XSDOT(IRUN,T,NF,NS,NXTJ,XF,XTRAJ,XDOT)
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ),XDOT(NS)
      COMMON/SNOIS/SDWSO(50),SDWS(10),SDWFO,SDWF
      COMMON/TRJCON/RE,G,OMEGA,RK1,RK2
      COMMON/TRAJ1/V,RLAT,RLON,ALT
      COMMON/TRAJ2/VE,VN,VU,FE,FN,FU,WE,WN,WU
      COMMON/TRAJ3/CEX,CEY,CEZ,CNX,CNY,CNZ,CUX,CUY,CUZ
      COMMON/TRAJ4/SINLAT,COSLAT,TANLAT
      COMMON/TRAJ5/OMEGAN,OMEGAU,RHOE,RHON,RHOZ,RKZ
      RLAT=XTRAJ(1)
      RLON=XTRAJ(2)
```

```
ALT=XTRAJ(4)
ROLL=XTRAJ(5)
PITCH=XTRAJ(6)
YAW=XTRAJ(7)
VE=XTRAJ(8)
VN=XTRAJ(9)
VU=XTRAJ(10)
FE=XTRAJ(11)
FN=XTRAJ(12)
FU=XTRAJ(13)
DROLL=XTRAJ(14)
DPITCH=XTRAJ(15)
DYAW=XTRAJ(16)
V=SQRT(VE**2+VN**2)
SINLAT=SIN(RLAT)
COSLAT=COS(RLAT)
TANLAT=SINLAT/COSLAT
SINLON=SIN(RLON)
COSLON=COS(RLON)
OMEGAN=OMEGA*COSLAT
OMEGAU=OMEGA*SINLAT
SX=SIN(YAW)
SY=SIN(ROLL)
SZ=SIN(PITCH)
CX=COS(YAW)
CY=COS(ROLL)
CZ=COS(PITCH)
RHOE=-VN/RE
RHON=VE/RE
RHOU=VE*TANLAT/RE
WE=RHOE
WN=RHON + OMEGAN
WU=RHOU + OMEGAU
RKZ=VU/RE
F42=2.0*(OMEGAN*VN + OMEGAU*VU) + RHON*VN/COSLAT**2
F43=RHOU*RHOE + RHON*RKZ
F44=-RHOE*TANLAT - RKZ
F52=-2.0*OMEGAN8VE - RHON*VE/COSLAT**2
F53=RHON*RHOU - RHOE*RKZ
F63=2.0*G/RE - RHON**2 - RHOE**2
F92=WN + RHOU*TANLAT
CXE=SX
CXN=SZ*CY
CXU=CZ*CY
CYE=-CY*SX
CYN=SZ*SY*SX + CZ*CX
CYU=CZ*SY*SX - SZ*CX
CZE=-CY*CX
CZN=-CZ*SX + SZ*SY*CX
CZU=CZ*SY*CX + SY*SX
CEX=CXE
CEY=CYE
```

```
      CEZ=CZE
      CNX=CXN
      CNY=CYN
      CZY=CZN
      CUX=CXU
      CUY=CYU
      CUZ=CZU
      FX=CXN*FN + CXE*FE + CXU*FU
      FY=CYN*FN + CYE*FE + CYU*FU
      FZ=CZN*FN + CZE*FE + CZU*FU
      WX=CXN*WN + CXE*WE + CXU*WU + DYAW
      WY=CYN*WN + CYE*WE + CYU*WU + DROLL
      WZ=CZN*WN + CZE*WE + CZU*WU + DPITCH
      WXP=0.0
      WXM=0.0
      WYP=0.0
      WYM=0.0
      WZP=0.0
      WZM=0.0
      IF (WX .GE. 0.0 ) WXP=WX
      IF (WX .LE. 0.0 ) WXM=WX
      IF (WY .GE. 0.0 ) WYP=WX
      IF (WY .LE. 0.0 ) WYM=WX
      IF (WZ .GE. 0.0 ) WZP=WZ
      IF (WZ .LE. 0.0 ) WZM=WZ
      XDOT(1)=XS(2)*RHOU/COSLAT - XS(3)*RHON/(RE*COSLAT)
     1       + XS(4)/(RE*COSLAT)
      XDOT(2)=XS(3)*RHOE/RE + XS(5)/RE
      XDOT(3)=XS(6)
      XDOT(4)=XS(2)*F42 + XS(3)*F43 + XS(4)*F44
     1       + XS(5)*(WU + OMEGAU) - XS(6)*(WN + OMEGAN)
     1       - XS(8)*FU + XS(9)*FN
     1       + XS(34)*CEX + XS(35)*CEY + XS(36)*CEZ
     1       + XS(37)*CEX*FX + XS(38)*CEY*FY
     1       + XS(39)*CEZ*FZ - XS(40)*CEX*FZ
     1       + XS(41)*CEX*FY + XS(42)*CEY*FZ
     1       - XS(43)*CEY*FX - XS(44)*CEZ*FY
     1       + XS(45)*CEZ*FX + XS(48)
      XDOT(5)=XS(2)*F52 + XS(3)*F53 - XS(4)*2.0*WU - XS(5)*RKZ
     1       + XS(6)*RHOE + XS(7)*FU - XS(9)*FE
     1       + XS(34)*CNX + XS(35)*CNY + XS(36)*CNZ
     1       + XS(37)*CNX*FX + XS(38)*CNY*FY
     1       + XS(39)*CNZ*FZ - XS(40)*CNX*FZ
     1       + XS(41)*CNX*FY + XS(42)*CNY*FZ
     1       - XS(43)*CNY*FX - XS(44)*CNZ*FY
     1       + XS(45)*CNZ*FX + XS(49)
      XDOT(6)= - XS(2)*2.0*OMEGAU*VE + XS(3)*(F63 - RK1)
     1       + XS(4)*2.0*WN - XS(5)*2.0*RHOE
     1       - XS(6)*RK2 - XS(7)*FN
     1       + XS(8)*FE + XS(34)*CUX + XS(35)*CUY
     1       + XS(36)*CUZ + XS(37)*CUZ*FX + XS(38)*CUY*FY
     1       + XS(39)*CUZ*FZ - XS(40)*CUX*FZ + XS(41)*CUX*FY
     1          + XS(42)*CUY*FZ - XS(43)*CUY*FX
     1          - XS(44)CUZ*FY + XS(45)*CUZ*FX
     1          + XS(46)*(RK1 - RK2 + V/DALT)
     1          + XS(47)*(RK1*ALT + RK2*VU)
     1          + XS(50)
      XDOT(7)= - XS(3)*RHOE/RE - XS(5)/RE + XS(8)*WU
     1          - XS(9)*WN + XS(10)*CEX + XS(11)*CEY
     1          + XS(12)*CEZ + XS(13)*CEX*FX
     1          + XS(14)*CEX*FZ + XS(15)*CEY*FY
     1          + XS(16)*CEY*FZ + XS(17)*CEZ*FZ
     1          - XS(18)*CEZ*FY - XS(19)*CEX*FX*FY
     1          + XS(20)*CEY*FX*FZ - XS(21)*CEZ*FX*FY
     1          + XS(22)*CEX*WXP + XS(23)*CEX*WXM
     1          + XS(24)*CEY*WYP + XS(25)*CEY*WYM
     1          + XS(26)*CEZ*WZP + XS(27)*CEZ*WZM
     1          + XS(28)*CEX*WZ  - XS(29)*CEX*WY
     1          - XS(30)*CEY*WZ + XS(31)*CEY*WX
     1          + XS(32)*CEZ*WY - XS(33)*CEZ*WX
      XDOT(8)= - XS(2)*OMEGAU -XS(3)*RHON/RE + XS(4)/RE
     1          - XS(7)*WU + XS(9)*WE + XS(10)*CNX
     1           + XS(11)*CNY + XS(12)*CNZ
     1           + XS(13)*CNX*FX + XS(14)*CNX*FZ
     1          + XS(15)CNY*FY + XS(16)*CNY*FZ
     1          + XS(17)*CNZ*FZ - XS(18)*CNZ*FY
     1          - XS(19)*CNX*FY*FZ + XS(20)*CNY*FX*FZ
     1          - XS(21)*CNZ*FX*FY + XS(22)*CNX*WXP
     1          + XS(23)*CNX*WXM + XS(26)*CNZ*WZP
     1          + XS(27)*CNZ*WZM +XS(28)*CNX*WZ
     1          - XS(29)*CNX*WY - XS(30)*CNY*WZ
     1          + XS(31)*CNY*WX + XS(32)*CNZ*WY
     1          - XS(33)*CNZ*WX
      XDOT(9)= XS(2)*F92 - XS(3)*RHOU/RE + XS(4)*TANLAT/RE
     1          + XS(7)*WN - XS(8)*WE + XS(10)*CUX + XS(11)*CUY
     1          + XS(12)*CUZ + XS(13)*CUX*FX + XS(14)*CUX*FZ
     1          + XS(15)*CUY*FY + XS(16)*CUY*FZ
     1          + XS(17)*CUZ*FZ - XS(18)*CUZ*FY
     1          - XS(19)*CUX*FY*FZ + XS(20)*CUY*FX*FZ
     1          - XS(21)*CUZ*FX*FY + XS(22)*CUX*WXP
     1          + XS(23)*CUX*WXM + XS(24)*CUY*WYP
     1          + XS(25)*CUY*WYM + XS(26)*CUZ*WZP
     1          + XS(27)*CUZ*WZM + XS(28)*CUX*WZ
     1          - XS(29)*CUX*WY - XS(30)*CUY*WZ
     1          + XS(31)*CUY*WX + XS(32)*CUZ*WY
     1          - XS(33)*CUZ*WX
      DO 10 I=10,45
   10 XDOT(I)=0.0
      XDOT(46)= - XS(46)*V/DALT
      XDOT(47)=0.0
      XDOT(48)= - XS(48)*V/DGE
      XDOT(49)= - XS(49)*V/DGN
      XDOT(50)= - XS(50)*V/DGU
      RETURN
      ENTRY XSDOTO
      ALPHA=XTRAJ(3)
      DO 25 I=1,6
   25 XS(I)=GAUSS(0.0,SDWSO(I))
      DO 45 I=10,50
   45 XS(I)=GAUSS(0.0,SDWSO(I))
      RLAT=XTRAJ(1)
      OMEGA=7.2921151E-5
      OMEGAN=OMEGA*COS(RLAT)
      OMEGAU=OMEGA*SIN(RLAT)
      RK1=3E-2
      RK2=3E-4
      G=32.0881576
      RE=20925640.0
      RANHED=GAUSS(0.0,1.8138)
      CHEAD=COS(RANHEAD)
      SHEAD=SIN(RHEAD)
      CNX=-CHEAD
      CUX=SHEAD
      SUY=CHEAD
      CNY=SHEAD
      WX=CNX*OMEGAN + CUX*OMEGAU
      WY=CUY*OMEGAU + CNY*OMEGAU
      XS(7)=XS(4)*2.*OMEGAU/G - XS(6)*2.*OMEGAN/G
     1        + XS(36)/G - XS(45) + XS(48)/G
      XS(8)=XS(5)*2.*OMEGAU/G - XS(6)*2.*OMEGAN/G
     1        + XS(36)/G - XS(45) + XS(48)/G
      XS(9)=( - XS(5)/RE + XS(8)*OMEGAU + XS(12)
     1        + XS(31)*CNY*WX)/OMEGAU
      RETURN
      END

SUBROUTINE TRAJO(IRUN,T,NF,NS,NXTJ,XF,XS,XTRAJ)
      DIMENSION XF(NF),XS(NS),XTRAJ(NXTJ)
      COMMON/TRJCON/RE,G,OMEGA,RK1,RK2
      INTEGER ITITLE(60),IPRSET(19),IRTSET(19)
      NAMELIST/PRDATA/NSEGT,TSTART,VTO,ROLLO,PITCHO,HEADO,ALFAO
     1     GLATO,TLONO,ALTO,IPRNT,IRITE,IPLOT,ROLRAT,ROLTC,
     1     LLMECH,LUNIT,RELERR,ABSERR,IPRSET,IRTSET
C    READ AND ECHO TITLE AND INPUT DATA FROM PROFGEN (TAPE 3)
      READ(3) ITITLE,TODAY,CLOCK
      READ(3) NSEGT,TSTART,VTO,ROLLO,PITCHO,HEADO,ALFAO
     1        GLATO,TLONO,ALTO,IPRNT,IRITE,IPLOT,ROLRAT,
     1        ROLTC,LLMECH,LUNIT,RELERR,ABSERR,IPRSET,IRTSET
      DO 10 I=1,16
   10 READ(3) DUM
      IF (IRUN .NE. 1) RETURN
      WRITE(6,PRDATA)
      WRITE(6,100)(ITITLE(I),I=1,6),TODAY,CLOCK
      OMEGA=70292115E-5
      G=32.12698510
      RE=20925640.0
```

```
    RK1=3.0E-2
    RK2=3.0E-4
100 FORMAT (//5X,"THE ABOVE DATA WAS USED TO CREATE THE TRAJECTORY
   *USING 'PROFGEN':"
   *      //10X,"TRAJECTORY TITLE:  ",6A10,
   *      /10X,TRAJECTORY RUN DATE AND TIME: "A10,5X,A10)
    RETURN
    END

    SUBROUTINE CONVRT(NWSO,NWS,SDWSO,SDWS)
    DIMENSION SDWSO(NWSO),SDWS(NWS)
    PI=ABS(ACOS(-1.0))
C    DEGREES TO RADIANS CONVERSION
    CONV1=PI/180.0
C    HOURS TO SECONDS CONVERSION
    CONV2=3600.0
C    ARCMINUTES TO RADIANS CONVERTION
    CONV3=(1.0/60.0)*CONV1
C    ARCSECONDS TO RADIANS CONVERSION
    CONV4=(1.0/60.0)*CONV3
C    G'S TO FEET PER SECOND SQUARED CONVERSION
    CONV5=32.2
C    MICRO G'S TO FEET PER SECOND SQUARED CONVERSION
     CONV6=32.2E-6
C     DEGREES PER HOUR TO RADIANS PER SECOND CONVERSION
     CONV7=CONV1/CONV2
C     PARTS PER MILLION CONVERSION
     CONV8=1.0E-6
C     MICRORADIANS TO RADIANS CONVERSION
     CONV9=1.0E-6
C     G SQUARED TO (FEET PER SECOND SQUARED) CONVERSION
     CONV10=32.2**2
C     SQUARE ROOT HOUR TO SQUARE ROOT SECOND CONVERSION
     CONV11=60.0
C     NAUTICAL MILES TO FEET CONVERSION
     CONV12=6076.10333
     DO 15 I=1,2
 15  SDWSO(I)=SDWSO(I)*CONV3
     DO 20 I=10,12
 20  SDWSO(I)=SDWSO(I)*CONV7
     DO 40 I=13,18
 40  SDWSO(I)=SDWSO(I)*CONV7/CONV5
     DO 50 I=19,21
 50  SDWSO(I)=SDWSO(I)*CONV7/CONV5**2
     DO 60 I=22,27
 60  SDCWSO(I)=SDWSO(I)*CONV8
     DO 70 I=28,33
 70  SDWSO(I)=SDWSO(I)*CONV4
     DO 80 I=34,36
 80  SDWSO(I)=SDWSO(I)*CONV6
     DO 90 I=37,39
 90  SDWSO(I)=SDWSO(I)*CONV8
```

```
      DO 100 I=40,45
100   SDWSO(I)=SDWSO(I0*CONV6
      DO 110 I=48,50
110   SDWSO(I)=SDWSO(I)*CONV6
      DO 120 I=2,4
120   SDWS(I)=SDWS(I)*CONV1/(CONV2*CONV11)
      DO 130 I=5,7
130   SDWS(I)=SDWS(I)*CONV6/CONV11
      DO 140 I=8,10
140   SDWS(I)=SDWS(I)*CONV6
      RETURN
      END
```

VITA

Captain Donald J. Kocian was born in Winner, South Dakota in 1954. He graduated from high school in Spencer Nebraska and, subsequently, entered the U. S. Air Force Academy where he majored in Electrical Engineering. Upon graduation and commissioning, he completed navigation and electronic warfare trainng. He was assigned to Ellsworth Air Force Base, South Dakota, where he was a B-52 electronic warfare officer and flight instructor. Capt Kocian entered the Air Force Institute of Technology in August 1982 in the inertial navigation and stochastic estimation and control sequences. He is a staff officer at Strategic Air Command Headquaters at Offutt Air Force Base, Nebraska.

Permanent address: Route 2
Spencer, NE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A172420

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GE/ENG/86J-2

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL (If applicable)
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State and ZIP Code)
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 454333

7b. ADDRESS (City, State and ZIP Code)

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Avionics Laboratory

8b. OFFICE SYMBOL (If applicable)
AFWAL/AAAN-2

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State and ZIP Code)

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE (Include Security Classification)
See Box 19

12. PERSONAL AUTHOR(S)
Donald J. Kocian, B.S.E.E., Capt, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT (Yr., Mo., Day)
1986 Jun

15. PAGE COUNT
122

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.
17	07	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Inertial Navigation, Vibration,
Mathematical Models

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: STUDY OF THE EFFECTS OF VIBRATION ON INERTIAL NAVIGATION SYSTEM ACCURACY

Approved for public release: IAW AFR 190-1.
Lynn Wolaver 3 Sept 86
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (A[illegible])
Wright-Patterson AFB OH 45433

Thesis Chairman: Robert Fontana
Professor Emeritus

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Robert E. Fontana, Professor Emeritus

22b. TELEPHONE NUMBER (Include Area Code)
513-255-3576

22c. OFFICE SYMBOL
AFIT/ENG

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

This study examines the effects of airframe vibration on the accuracy of a strapdown inertial navigation system. A stochastic model of the system error equations is included, as are two models of airframe vibration. Software subroutines for model implementation in SOFE are included.

A representative C-130A flight profile was developed using a flight trajectory generator, PROFGEN. The system errors induced in the inertial navigation system by simulating this mission are included as are those caused by vibration. Vibration induced errors were found to be very small and orders of magnitude smaller than those caused by other error sources.